Lab 11 & 12 Image Processing II

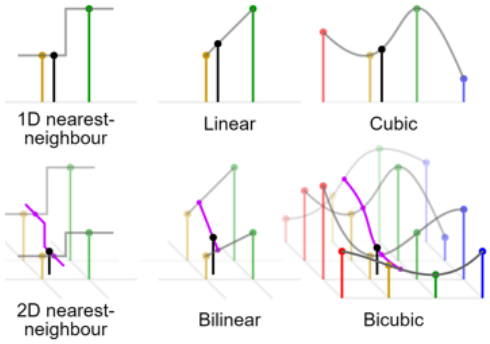
Elena Montalvo
Data collected with Darren Ray
ECE3101L - Andrew Pagnon
December 10, 2022

Summary

Lab 11

Resize an image

Resizing images is really important it allows us to process and store different data however this could lead to issues because by shrinking a photo or a file we lose a fraction of the quality. Deeping on how much we resize it we could see drastic changes that even if the image is reconstructed, many of the pixels are lost in the process. We are going to show how the images get affected when shrunked by ½,¼ ⅛, 1/16, 1/32, 1/64, and 1/128, we are going it shows the difference to the nearest pixel, bilinear and bicubic



Filter an image

We could also use different types of filters to analyze images

Low pass Filtering
Write a program to perform low pass filtering of one or more of the images provided.
You can test your program at the two different masks H given below.

$H = \frac{1}{9}[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ $H = \frac{1}{16}[1\ 2\ 1\ 2\ 4\ 2\ 1\ 2\ 1]$

High pass filtering
Write a program to perform highpass filtering of one or more of the images provided.
You can test your program at the two different masks H given below.

$H = [-1\ -1\ -1\ -1\ 8\ -1\ -1\ -1\ -1]$ $H = [0\ -1\ 0\ -1\ 4\ -1\ 0\ -1\ 0]$

Enhanced Image using High pass filtering
Write a program to generate an enhanced image using high pass filtering of one or more of the images provided.
You can test your program with the two high pass filter masks H given above.

$EnhancedImage = OriginalImage + HighPassHMaskFilteredImage$

Spatial Filtering
Write a program to perform spatial filtering of one or more of the images provided.
You can test your program with the masks H given below

$H = [1\ 1\ 1\ 1\ -8\ 1\ 1\ 1\ 1]$ $SpacialFilteredImage = OriginalImage - HMaskFilteredImage$

Enhancement Using the Laplacian
Write a program developed to implement the Laplacian enhancement technique.
You can test your program with the first high pass masks H given above

$LaplacianFilteredImage = OriginalImage + FirstHighPassHMaskFilteredImage$

Lab 12

Enhance contrast using histogram equalization

'histeq' enhances the contrast of images by transforming the values in an intensity image, or the values in the color map of an indexed image, so that the histogram of the output image approximately matches a specified histogram

Two-dimensional discrete Fourier transform

We are going to use the discrete Fourier to analysis the Two-dimensional scope of the image and compute the average values

$$F(0,0) = \frac{1}{MN} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(x,y)$$

Lowpass and Highpass Filtering using Gaussian

Low pass Filtering using Gaussian
Implement the Gaussian low pass filter.
You must be able to specify the size, M x N, of the resulting 2D function.
In addition, you must be able to specify where the 2D location of the center of the Gaussian function.
High pass Filtering Using a Low pass Gaussian filtered Image
Subtract your low pass filtered image from the original to obtain a sharpened image.
You will note that the resulting image does not resemble the Gaussian high pass results.
Adjust the variance (sigma) of your Gaussian low pass filter until the result is obtained.

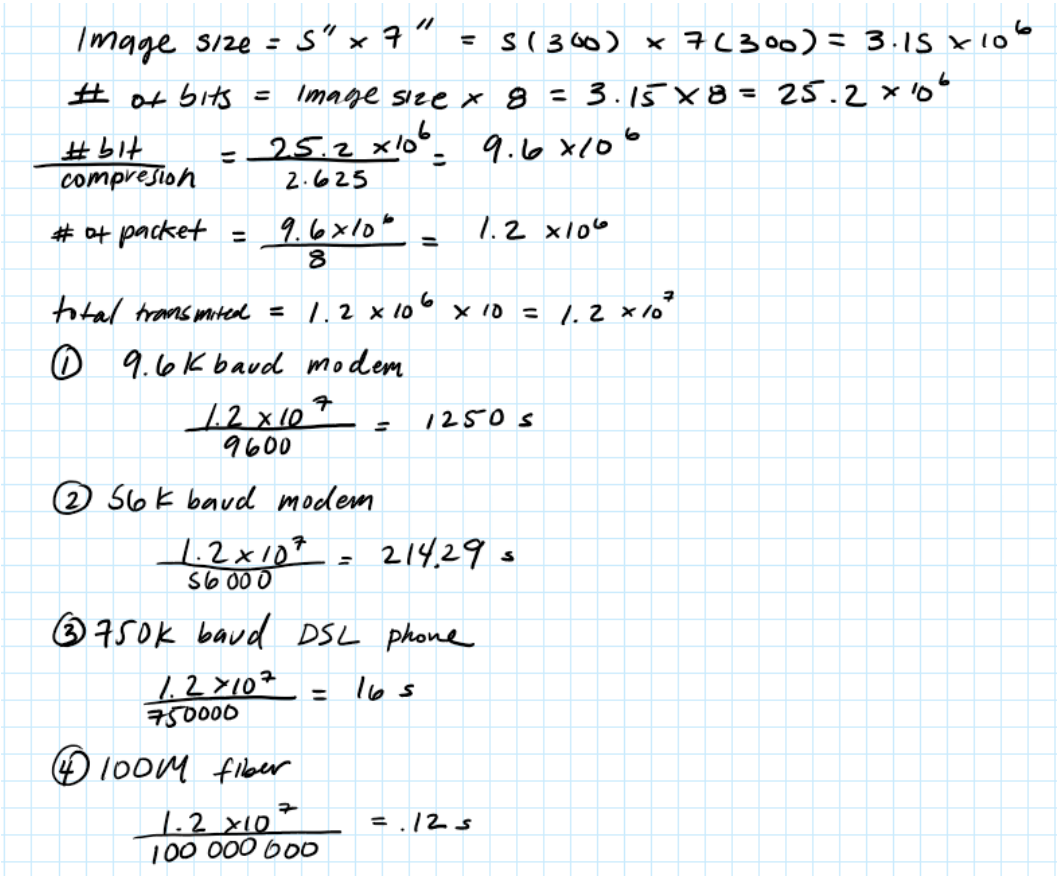
Objective

1. Read and write images in MATLAB
2. Change the quantization and size of an image in MATLAB
3. Design two-space dimensional low pass and high pass filters for image enhancement in MATLAB.
4. Equalize an image in Matlab using histograms
5. Compute a centered image Fourier spectrum
6. Design two-dimensional Gaussian low/high pass filters in MATLAB.

7. Adjust the variance of the Gaussian low/high pass filter in MATLAB.

Prelab

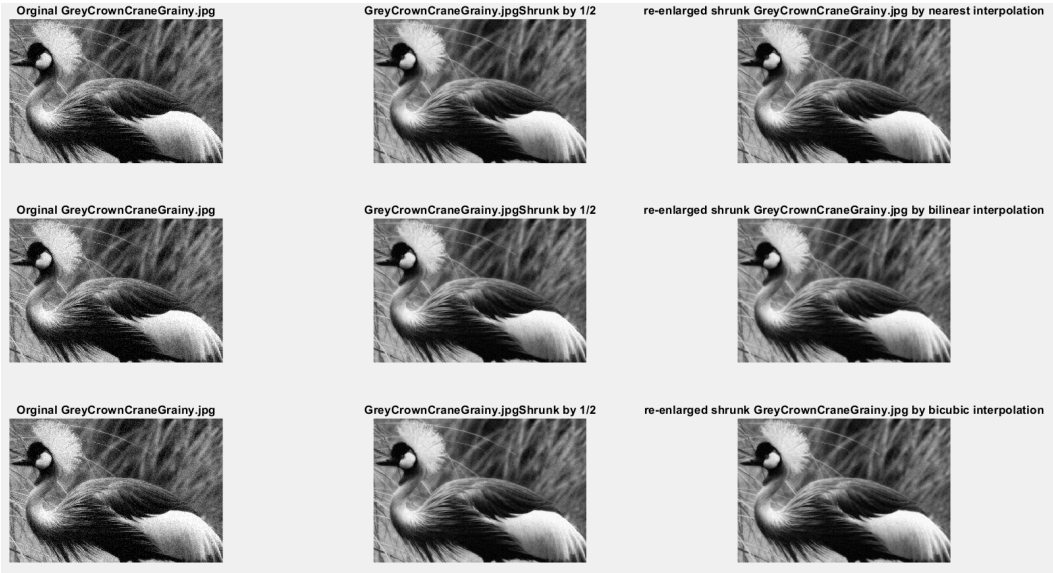
A standard measure of transmission for digital data is the baud rate, defined as the number of bits transmitted per second. Generally, packet transmission consists of a start bit, a byte of information, and a stop bit.
How long would it take to transmit a 5”x7” picture with a resolution of 300 pixels/inch with 256 levels per pixel and a compression ratio of 2.625 using a:

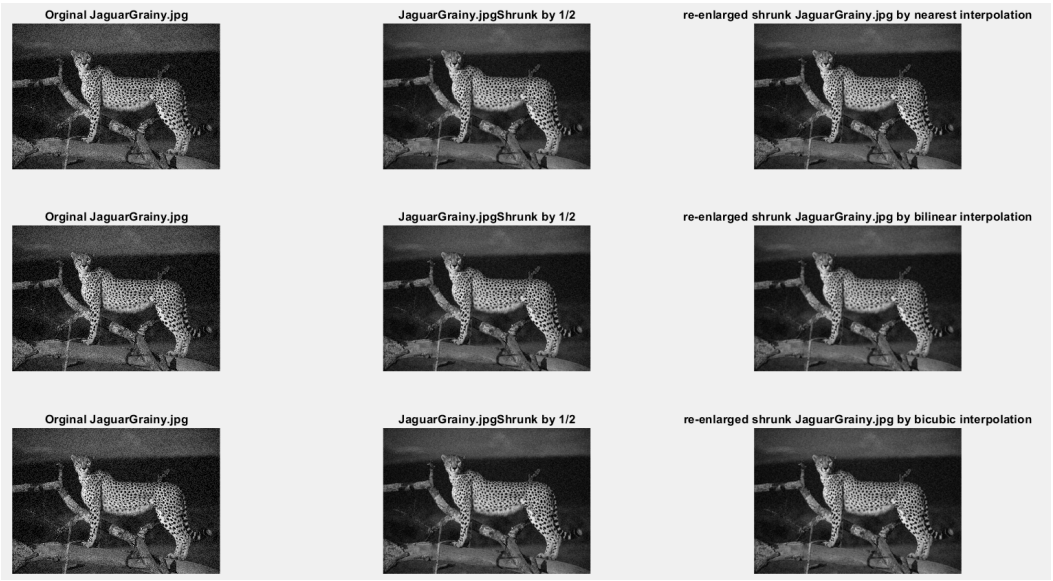


Results

Lab 11 part 1 results

Shrunk by ½





Orginal moonImage.tif

moonImage.tifShrunk by 1/2

re-enlarged shrunk moonImage.tif by nearest interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/2

re-enlarged shrunk moonImage.tif by bilinear interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/2

re-enlarged shrunk moonImage.tif by bicubic interpolation

Shrunk by ¼

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/4

re-enlarged shrunk GreyCrownCraneGrainy.jpg by nearest interpolation

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/4

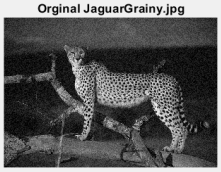
re-enlarged shrunk GreyCrownCraneGrainy.jpg by bilinear interpolation

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/4

re-enlarged shrunk GreyCrownCraneGrainy.jpg by bicubic interpolation

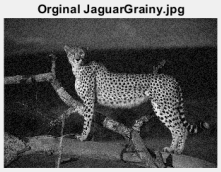
Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/4

re-enlarged shrunk JaguarGrainy.jpg by nearest interpolation

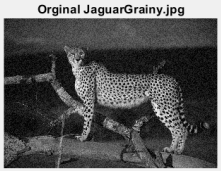
Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/4

re-enlarged shrunk JaguarGrainy.jpg by bilinear interpolation

Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/4

re-enlarged shrunk JaguarGrainy.jpg by bicubic interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/4

re-enlarged shrunk moonImage.tif by nearest interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/4

re-enlarged shrunk moonImage.tif by bilinear interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/4

re-enlarged shrunk moonImage.tif by bicubic interpolation

Shrunk by ⅛

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/8

re-enlarged shrunk GreyCrownCraneGrainy.jpg by nearest interpolation

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/8

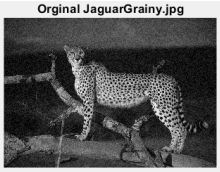
re-enlarged shrunk GreyCrownCraneGrainy.jpg by bilinear interpolation

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/8

re-enlarged shrunk GreyCrownCraneGrainy.jpg by bicubic interpolation

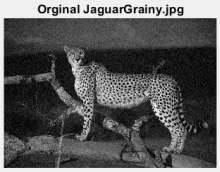
Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/8

re-enlarged shrunk JaguarGrainy.jpg by nearest interpolation

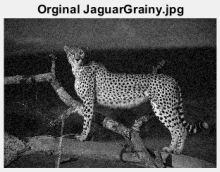
Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/8

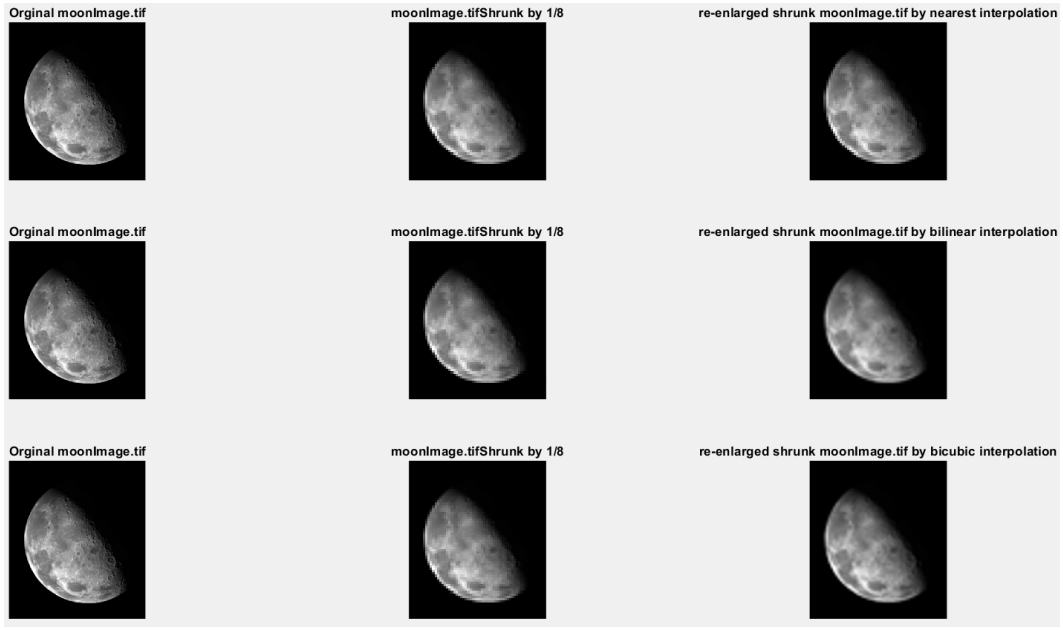
re-enlarged shrunk JaguarGrainy.jpg by bilinear interpolation

Orginal JaguarGrainy.jpg

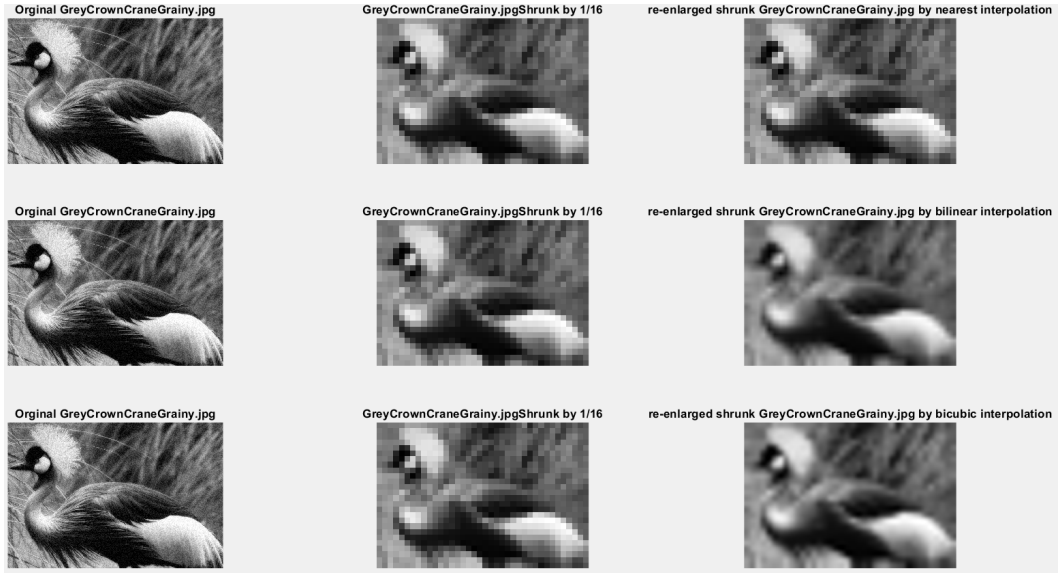


JaguarGrainy.jpgShrunk by 1/8

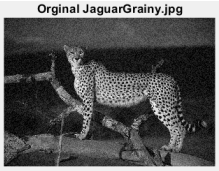
re-enlarged shrunk JaguarGrainy.jpg by bicubic interpolation



Shrunk by 1/16

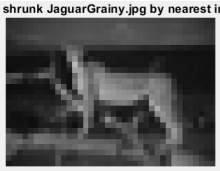


Orginal JaguarGrainy.jpg

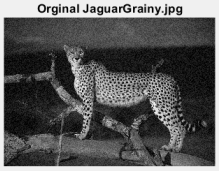


JaguarGrainy.jpgShrunk by 1/16

re-enlarged shrunk JaguarGrainy.jpg by nearest interpolation



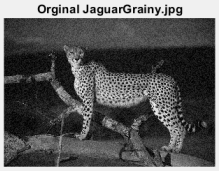
Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/16

re-enlarged shrunk JaguarGrainy.jpg by bilinear interpolation

Orginal JaguarGrainy.jpg



JaguarGrainy.jpgShrunk by 1/16

re-enlarged shrunk JaguarGrainy.jpg by bicubic interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/16

re-enlarged shrunk moonImage.tif by nearest interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/16

re-enlarged shrunk moonImage.tif by bilinear interpolation

Orginal moonImage.tif

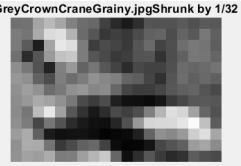
moonImage.tifShrunk by 1/16

re-enlarged shrunk moonImage.tif by bicubic interpolation

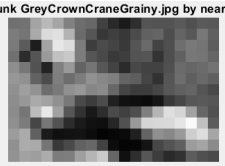
Shrunk by 1/32

Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/32

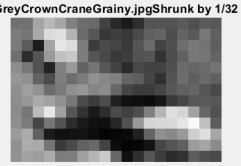


re-enlarged shrunk GreyCrownCraneGrainy.jpg by nearest interpolation



Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/32



re-enlarged shrunk GreyCrownCraneGrainy.jpg by bilinear interpolation

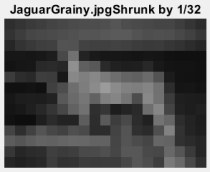
Orginal GreyCrownCraneGrainy.jpg

GreyCrownCraneGrainy.jpgShrunk by 1/32

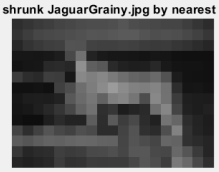
re-enlarged shrunk GreyCrownCraneGrainy.jpg by bicubic interpolation

Orginal JaguarGrainy.jpg

JaguarGrainy.jpgShrunk by 1/32

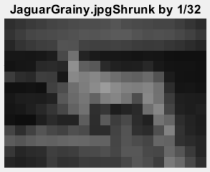


re-enlarged shrunk JaguarGrainy.jpg by nearest interpolation



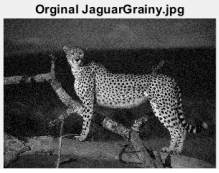
Orginal JaguarGrainy.jpg

JaguarGrainy.jpgShrunk by 1/32

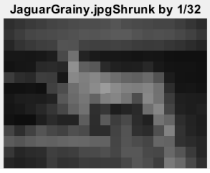


re-enlarged shrunk JaguarGrainy.jpg by bilinear interpolation

Orginal JaguarGrainy.jpg



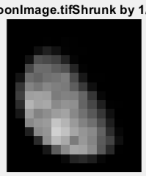
JaguarGrainy.jpgShrunk by 1/32



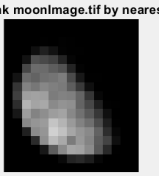
re-enlarged shrunk JaguarGrainy.jpg by bicubic interpolation

Orginal moonImage.tif

moonImage.tifShrunk by 1/32

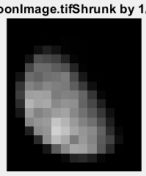


re-enlarged shrunk moonImage.tif by nearest interpolation



Orginal moonImage.tif

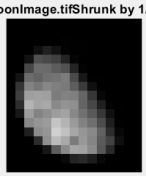
moonImage.tifShrunk by 1/32



re-enlarged shrunk moonImage.tif by bilinear interpolation

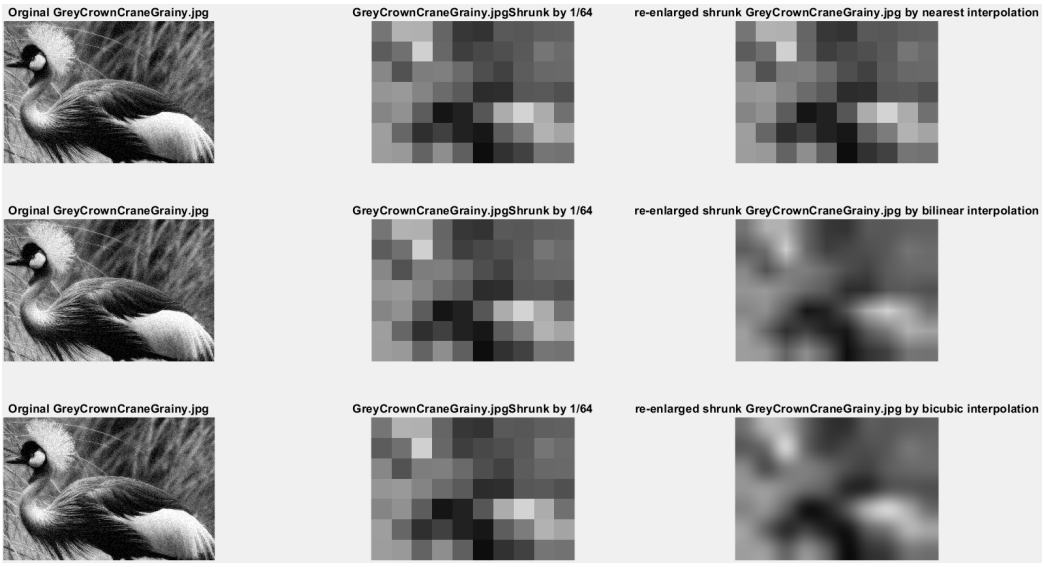
Orginal moonImage.tif

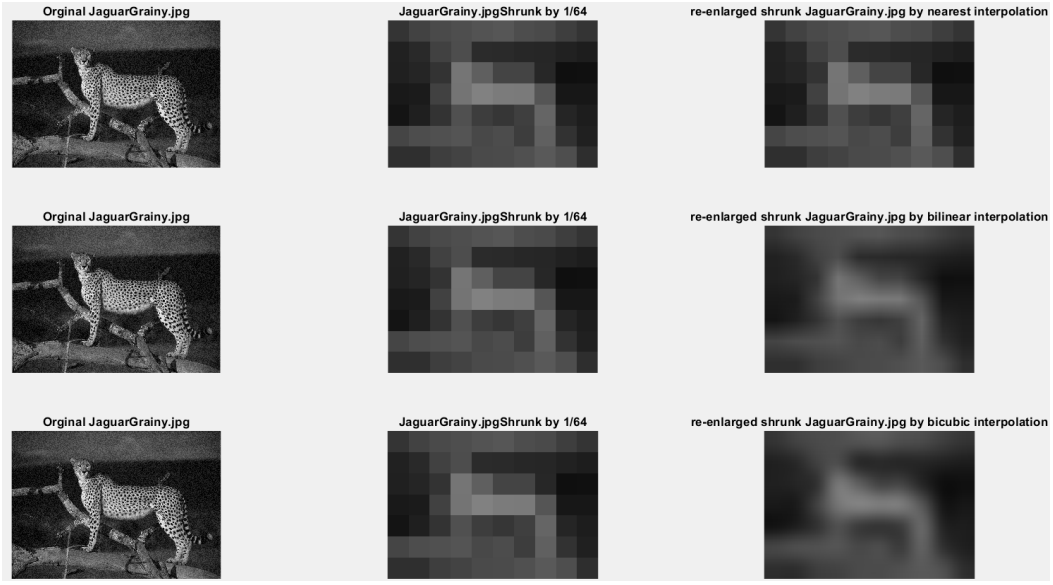
moonImage.tifShrunk by 1/32

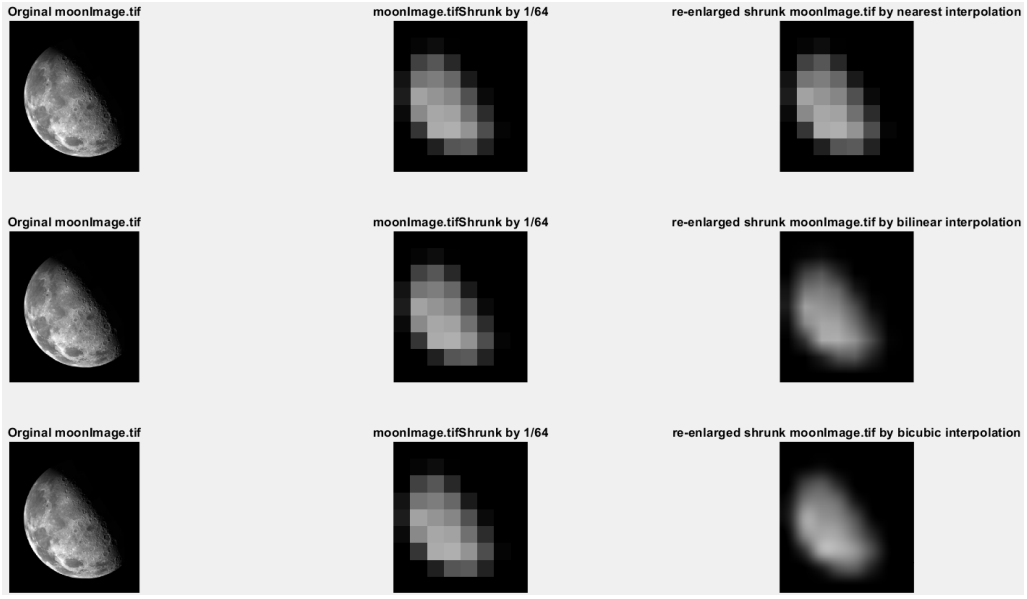


re-enlarged shrunk moonImage.tif by bicubic interpolation

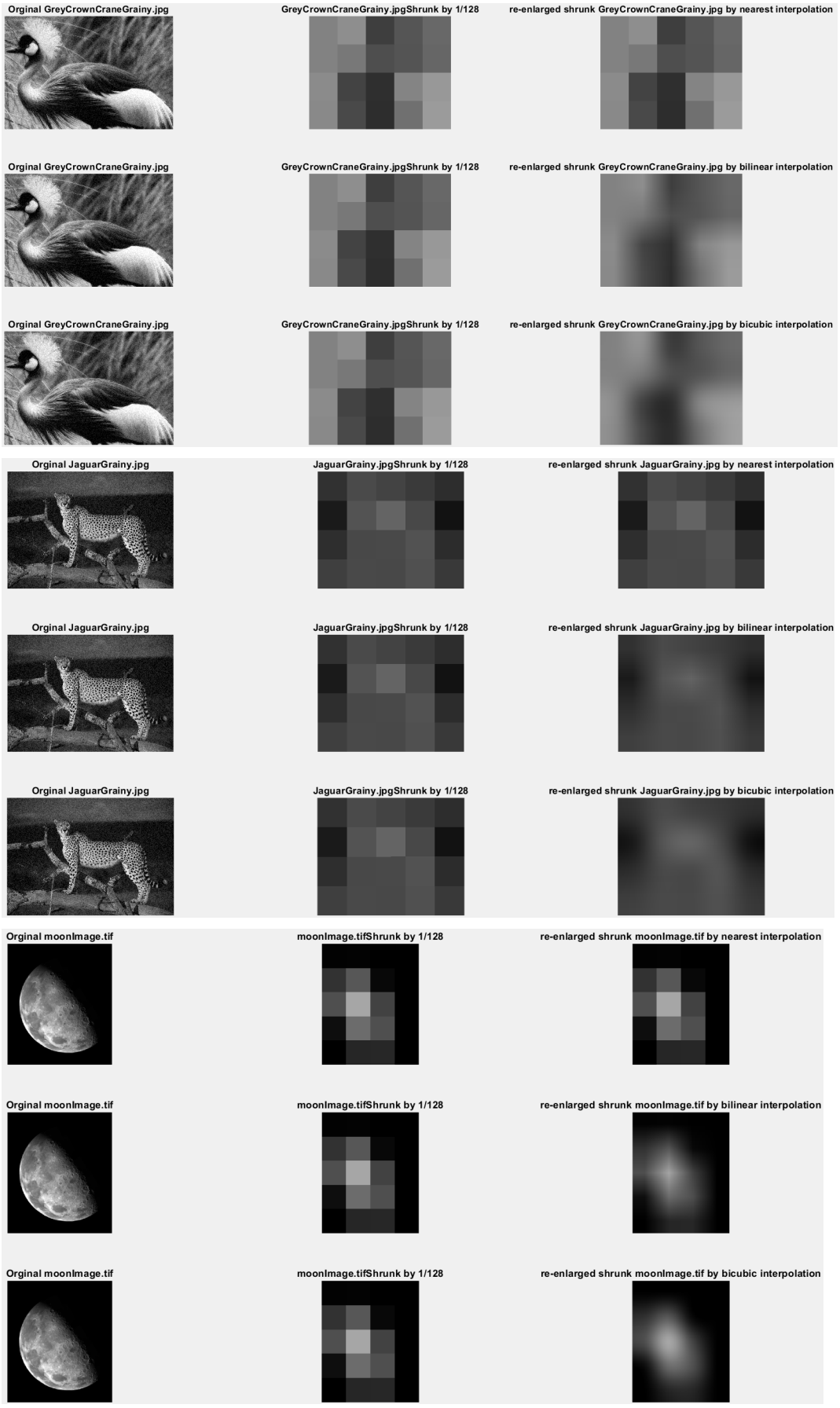
Shrunk by 1/64







Shrunk by 1/128



Lab 11 part 2 results

Original Image

GreyCrownCrane.jpg LPF1

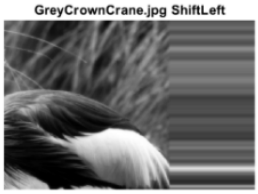
GreyCrownCrane.jpg LPF2

GreyCrownCrane.jpg Spatial

GreyCrownCrane.jpg HPEnhanced1

GreyCrownCrane.jpg HPEnhanced2

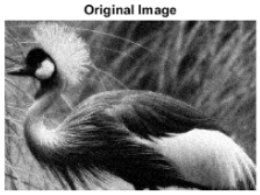
GreyCrownCrane.jpg ShiftLeft



GreyCrownCrane.jpg LaplacianScaled

GreyCrownCrane.jpg LPHPEnhanced2

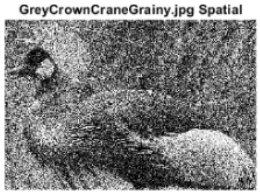
Original Image



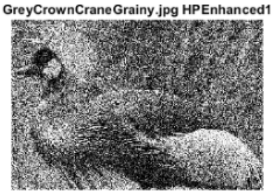
GreyCrownCraneGrainy.jpg LPF1

GreyCrownCraneGrainy.jpg LPF2

GreyCrownCraneGrainy.jpg Spatial



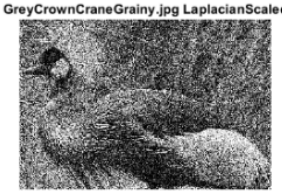
GreyCrownCraneGrainy.jpg HPEnhanced1



GreyCrownCraneGrainy.jpg HPEnhanced2

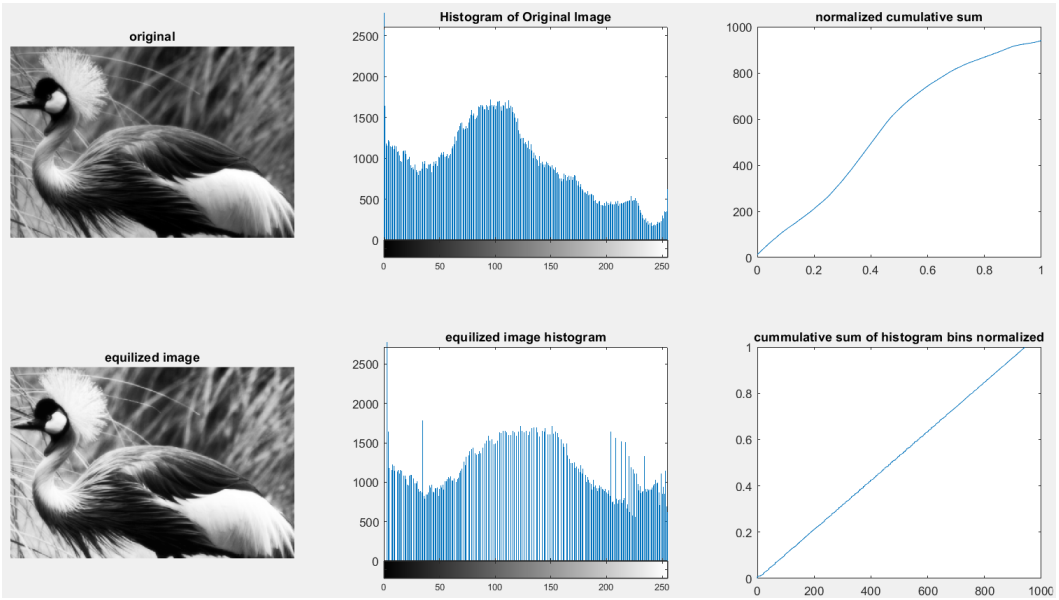
GreyCrownCraneGrainy.jpg ShiftLeft

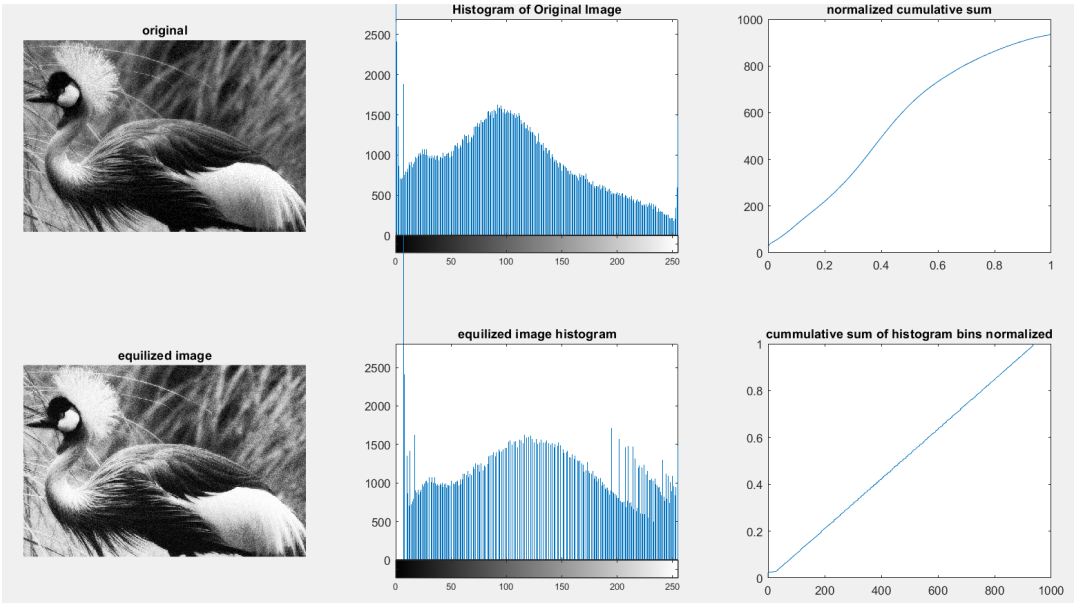
GreyCrownCraneGrainy.jpg LaplacianScaled



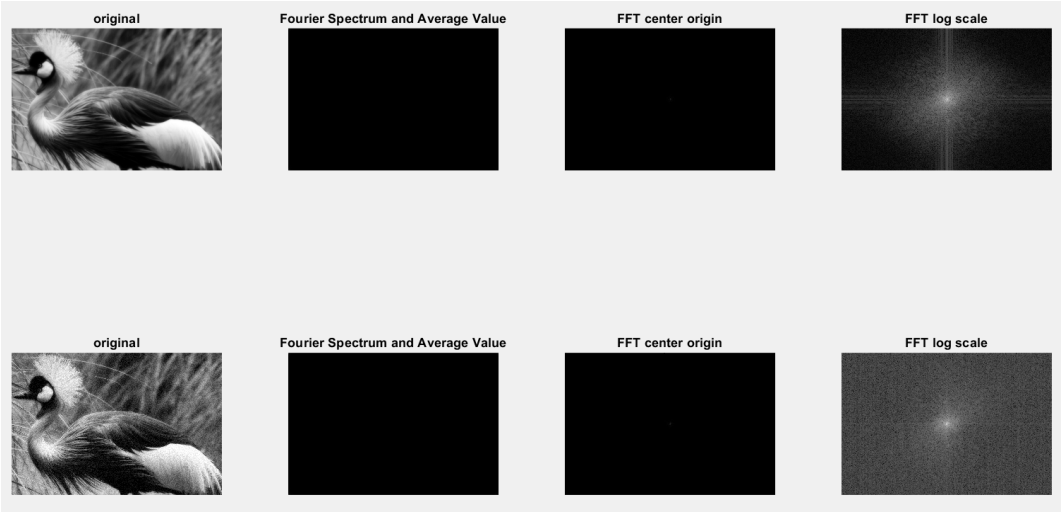
GreyCrownCraneGrainy.jpg LPHPEnhanced2

Lab 12 part 1 results

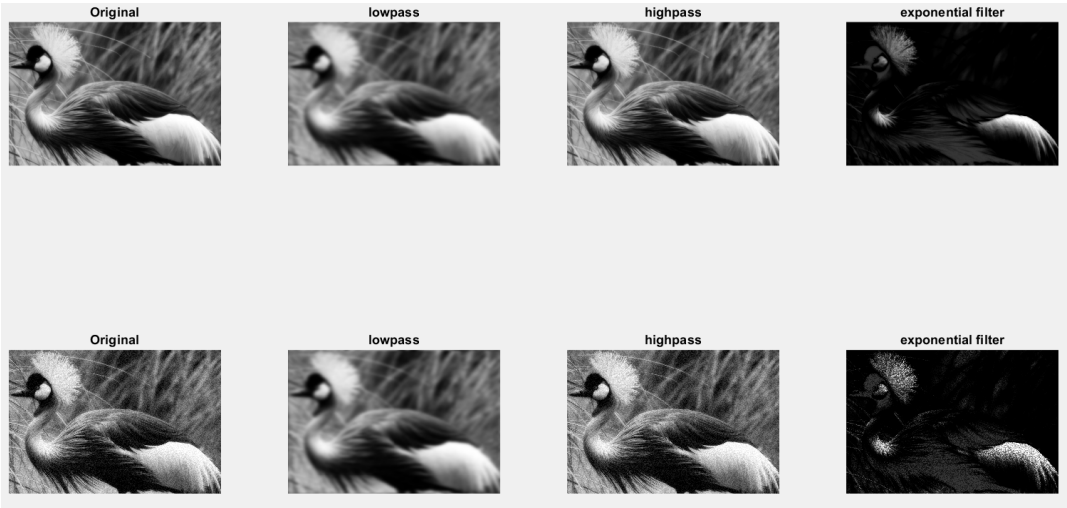




Lab 12 part 2 results



Lab 12 part 3 results



Conclusion

In this lab we learned different ways to process an image and we understood each of the following analysis limitations.
When we shrunk the images to a greater factor the image to worst and worst until it became impossible to distinguish due to the significant loss of pixels. 1/16 or 1/ (2)^4 the image was harder to recognize. we also saw that when we tried to re-enlarge the image there were many differences between the nearest, the bilinear, and the bicubic. We notice that the nearest was just each block however the bilinear and bicubic tend to look more blended because this type of reconstruction took more consideration of other pixels to recreate the image.
We also saw the different types of filters affected the quality of the images the Low pass filter only allows low frequencies which is why the images than to be more smooth while the high

pass filter allows for high frequencies that why the images after passing through the high pass filter to look sharper.

We also enhanced the images by using ‘histeq’ which enhances the contrast of images by transforming the values in an intensity image, or the values in the color map of an indexed image so that the histogram of the output image approximately matches a specified histogram. Using the histogram we were able to see what color pixels than to be more prominent in the images.

Through the 2 Dimensional Fourier we learn about the importance of having the right scale because in the first two where we average then and center the Fourier transform we didn't see much of the affected in the images however once we change it to log from the colors were much clear and we were able to distinguish easier. the contrasts of the dark and lighter pixels in a 2-dimensional plane. I see it as looking into space, you have to have the right tools to see some stars or planets

Resources

Lab 11 Part 1 code

```
%Resizing an image by pixel replication
%Take an image, shrink it and re-enlarge the shrunk image back to its
%original size. Then plot the original, shrunk and re-enlarged versions
clear all; clc;close all;
%list of images to process
image_title_list = {'moonImage.tif','JaguarGrainy.jpg','GreyCrownCraneGrainy.jpg'};
%list of interpolation methods used to resize the image
%'nearest','bilinear','bicubic'
interpolation_method_list = {'nearest','bilinear','bicubic'};
%halved is the number of times the image is reduced by half
halved=7;
%the image is shrunk by 1/2^halved and re-enlarged by 2^halved
%for example for halved=3 where the image is halved 3 times
%the image is shrunk by 1/8 and re-enlarged by 8
for hlf=1:halved%loop once for every shrinking fraction which is 1/2^hlf
    for imgfls = 1:length(image_title_list)%loop once for every image
    %read the contents of each image into original_image_content
    %original_image_content is the content of each image file
    %image_title_list is a list of the title of each image file
        original_image_content = imread(image_title_list{imgfls});
         for intrplt = 1:length(interpolation_method_list)%loop once for every
interpolation method
            fig_num=hlf*length(interpolation_method_list)-(imgfls-1);
            fsf(fig_num)=figure(fig_num);%one full screen figure per "halved" value
            set(fsf(fig_num), 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
            subplot(length(interpolation_method_list),3,3*intrplt-2)%(r,c,p)rows,
columns,position
            %position is row=imgfls, column=1
            %plot the original image
            imshow(original_image_content);
            title(['Orginal ',image_title_list{imgfls}]);
            %shrink by 1/2^hlf
            shrunk_image=imresize(original_image_content,1/2^hlf);
            subplot(length(interpolation_method_list),3,3*intrplt-1)%(r,c,p)rows,
columns,position
            %position is row=imgfls, column=2
            %plot the shrunk image
            imshow(shrunk_image);
            title([image_title_list{imgfls},'Shrunk by 1/',num2str(2^hlf)]);
            %re-enlarge by 2^hlf back to the original image size
            %nearest-Nearest-neighbor interpolation; the output pixel is assigned the
            %value of the pixel that the point falls within. No other pixels are
considered
            %bilinear-Bilinear interpolation; the output pixel value is a weighted
average
            %of pixels in the nearest 2-by-2 neighborhood
            %reenlarged_image=imresize(shrunk_image,2^halved,'bilinear');
            %bicubic-Bicubic interpolation; the output pixel value is a weighted
average
```

```
            %of pixels in the nearest 4-by-4 neighborhood
            interpolation_method=interpolation_method_list{intrplt};
            reenlarged_image=imresize(shrunk_image,2^hlf,interpolation_method); %↙
arguments=(image, enlargement factor, interpolation method)
            subplot(length(interpolation_method_list),3,3*intrplt) %(r,c,p)rows,↙
columns,position
            %position is row=imgfls, column=3
            %plot the re-enlarged image
            imshow(reenlarged_image);
            title(['re-enlarged shrunk ',image_title_list{imgfls},' by ',↙
interpolation_method_list{intrplt},' interpolation']);
        end
    end
end
```

Lab 11 Part 2 code

```
%Image filtering
clear all; clc;close all
image_title_list = {'GreyCrownCrane.jpg','GreyCrownCraneGrainy.jpg'};
filter_name=↙
{'LPF1','LPF2','Spatial','HPEnhanced1','HPEnhanced2','ShiftLeft','LaplacianScaled','L↙
PHPEnhanced2'};
filter{1} = [1 1 1; 1 1 1; 1 1 1]/9;% Low Pass Filter 1
filter{2} = [1 2 1; 2 4 2; 1 2 1]/16;% Low Pass Filter 2
filter{3} = [1 1 1; 1 -8 1; 1 1 1];% Spatial Filter
filter{4} = [-1 -1 -1; -1 8 -1; -1 -1 -1];% High Pass Filter1
filter{5} = [0 -1 0; -1 4 -1; 0 -1 0];% High Pass Filter2
filter{6} = [0 0 0; 0 0 1; 0 0 0];% shift left
filter{7} = [-1 -1 -1; -1 8 -1; -1 -1 -1];% Laplacian for LaplacianScaled
filter{8} = [-1 -1 -1; -1 8 -1; -1 -1 -1];% High Pass Filter 2 for LPHPEnhanced2
for imgfls = 1:length(image_title_list)%loop for every image
    %read the contents of each image into original_image_content
    %original_image_content is the content of each image file
    %image_title_list is a list of the title of each image file
    original_image_content = im2double(imread(image_title_list{imgfls}));
    fsf(imgfls)=figure(imgfls);%create a full screen page for every image and its↙
filtered versions
    set(fsf(imgfls), 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
    subplot(3,3,1)%each figure has 3 rows and 3 columns with
    %top row = original and two low pass filtered
    %middle row = Spatial and two high pass filtered enhanced
    %lower row = Shifted left, Laplacian 0.9 scaled and High pass filtered
    %enhanced after low pass filtered
    imshow(original_image_content)%display the original image with no filter
    set(gcf, 'color', 'white');
    title('original image');
    for fltr = 1:length(filter_name)%process the original image using different↙
filter methods like
        %enhanced, scaled, low pass then high pass enhanced
        filtered_image = imfilter(original_image_content,filter{fltr}, 'replicate');
        %First filter the original image with one of the filter matrices listed above
        %Then every if statement will generate a processed image using the original↙
and/or
filtered images
        if strcmp(filtered_image{fltr}, 'LPF1') | strcmp(filtered_image{fltr},↙
'LPF2')
        processed_image = filtered_image;
        elseif strcmp(filter_name{fltr}, 'Spatial')
        processed_image = original_image_content-filtered_image;
        elseif strcmp(filter_name{fltr}, 'HPEnhanced1') | strcmp(filtered_image↙
{fltr}, 'HPEnhanced2')
        processed_image = original_image_content+filtered_image;
        elseif strcmp(filter_name{fltr}, 'ShiftLeft')
            for i=1:200 % number of pixels the image is shifted left
                filtered_image = imfilter(filtered_image,filtered_image↙
```

```
{fltr},'replicate');
            end
            processed_image=original_image_content;
        elseif strcmp(filter_name{fltr}, 'LaplacianScaled')
        processed_image = 0.9*(original_image_content+filtered_image);
        elseif strcmp(filter_name{fltr}, 'LPHPEnhanced2')
        processed_image = original_image_content+imfilter(imfilter
(original_image_content,filter{1},'replicate'),filter{fltr},'replicate');
        else
        processed_image = filtered_image;
        end
        subplot(3,3,fltr+1)%fltr+1 as the first image is the original outside of this
fltr loop
        imshow(processed_image)
        set(gcf, 'color', 'white');
        title([image_title_list{imgfls},' ',filtered_image{fl}]);
    end
end
```

Lab 12 Part 1 code

```
%Histogram Equalization
clear all; clc;close all;
imgfiles = {'GreyCrownCrane.jpg','GreyCrownCraneGrainy.jpg'};
for imgfls=1:length(imgfiles)
    [original_image{imgfls}] = imread(imgfiles{imgfls});
end
for imgfls = 1:length(imgfiles)
    original_img=original_image{imgfls};%avoid index inside the loop
    fsf(imgfls)=figure(imgfls);
    set(fsf(imgfls), 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
    subplot(2,3,1)
    imshow(original_img) % original image
    title('original');
    subplot(2,3,2)
    imhist(original_img);
    title('Histogram of Original Image');
    number_of_pixels=numel(original_img);
    imghist_orig_norm=imhist(original_img)./256 % Normalized
    cumulative_sum_orig=cumsum(imghist_orig_norm);
    x_axis=linspace(0,1,256);
    subplot(2,3,3)
    plot(x_axis,cumulative_sum_orig); %normalized cumulative sum
    title('normalized cumulative sum');
    equalized_image=histeq(original_img,number_of_pixels);
    subplot(2,3,4)
    imshow(equalized_image) % equilized image
    title('equilized image');
    subplot(2,3,5)
    imhist(equalized_image)
    title('equilized image histogram');
    imghist_eqlz_norm=imhist(equalized_image)./256; % Nomalized
    cumulative_sum_eqlz=cumsum(imghist_eqlz_norm);
    subplot(2,3,6)
    plot(cumulative_sum_eqlz,x_axis); %normalized cumulative sum
    title('cummulative sum of histogram bins normalized');
end
```

Lab 12 Part 2 code

```
clear all; clc;close all;
imgfiles={'GreyCrownCrane.jpg','GreyCrownCraneGrainy.jpg'};
for imgfls=1:length(imgfiles)
    [original_image{imgfls}] = imread(imgfiles{imgfls});
end
for imgfls = 1:length(imgfiles)
    original_img=original_image{imgfls};%avoid index inside the loop
    fsf(1)=figure(1);
    set(fsf(1), 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
    subplot(length(imgfiles),4,4*imgfls-3)
    imshow(original_img)
    title('original');
    % Fourier Spectrum and Average Value
    fft_img=fft2(im2double(original_img));
    abs_fft_img=abs(fft_img);
    subplot(length(imgfiles),4,4*imgfls-2)
    imshow(abs_fft_img,[]);
    title('Fourier Spectrum and Average Value');
    shift_fft_img=fftshift(abs_fft_img);
    abs_shift_fft_img=abs(shift_fft_img);
    subplot(length(imgfiles),4,4*imgfls-1)
    imshow(abs_shift_fft_img,[]);
    title('FFT center origin');
    log_abs_shift_fft_img=log(1+abs_shift_fft_img);
    subplot(length(imgfiles),4,4*imgfls)
    imshow(log_abs_shift_fft_img,[]);
    title('FFT log scale');
    % Average Value
    [M,N]=size(original_img);
    double_image=double(original_img);
    sum=0;
    for m=1:M
        for n=1:N
            sum=double_image(m,n)+sum;
        end
    end
    fft_average{imgfls}=sum/(M*N)
end
```

Lab 12 Part 3 code

```
%Filtering using Gaussian
clear all; clc;close all;
imgfiles={'GreyCrownCrane.jpg','GreyCrownCraneGrainy.jpg'};
sigmalp = 20;
sigmahp = 0.25;
for imgfls=1:length(imgfiles)
    [original_image{imgfls}] = imread(imgfiles{imgfls});
end
for imgfls = 1:length(imgfiles)
    original_img=original_image{imgfls};%avoid index inside the loop
    fsf(1)=figure(1);
    set(fsf(1), 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
    subplot(length(imgfiles),4,4*imgfls-3)
    imshow(original_img);
    title(['Original']);
    [M,N] = size(original_img);
    fft_img = fft2(im2double(original_img));
    u = 0:(M-1);
    v = 0:(N-1);
    idx = find(u>M/2);
    u(idx)= u(idx)-M;
    idy = find(v>N/2);
    v(idy)= v(idy)-N;
    [U,V] = meshgrid(v,u);
    D = sqrt(U.^2+V.^2);
    %Lowpass filtering using Gaussian
    Hlp = exp(-D.^2/(2*sigmalp.^2));
    Glp = fft_img.*Hlp;
    lowpass_filtered = real(ifft2(Glp));
    subplot(length(imgfiles),4,4*imgfls-2)
    imshow(lowpass_filtered,[ ])
    title(['lowpass']);
    %Highpass filtering form Lowpass filtered image
    Hhp = 1- exp(-D.^2/(2*sigmahp.^2));
    Ghp = fft_img.*Hhp;
    highpass_filtered = real(ifft2(Ghp));
    subplot(length(imgfiles),4,4*imgfls-1)
    imshow(highpass_filtered,[ ])
    title('highpass');
    highpass_filtered_dbl=double(highpass_filtered);
    exponential_filtered=highpass_filtered_dbl.^4;
    subplot(length(imgfiles),4,4*imgfls)
    imshow(exponential_filtered,[ ])
    title('exponential filter');
end
```