Table of Contents


```
%ECES435 Assignment 5 - By Wanyu Li and John Seitz
close all; clear all; clc;
%Note: Code works best if you run each section independently!
```

Part 1 - 1st Task - Contrast Enhancement

```
Imgs = {'imageCE1.tif','imageCE2.tif','imageCE3.tif','imageCE4.tif'};
%Load in images


for i = 1:length(Imgs) % Loop for all 4 images
    newimg = imread(Imgs{i});
    figure(i);
    subplot(1,2,1);
    imshow(newimg);
    title(['Image',sprintf('%d',i)]);
    subplot(1,2,2);
    imhist(newimg); % use imhist to calculate the image's PVH
    title('Image Pixel Histogram');
end
```

Image1

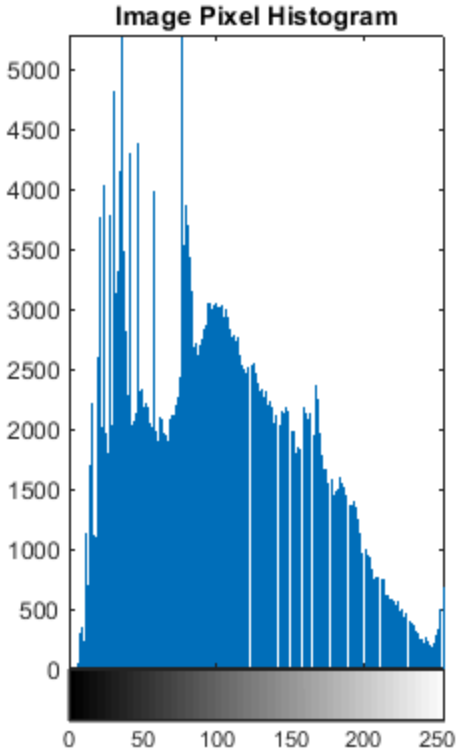


Image2

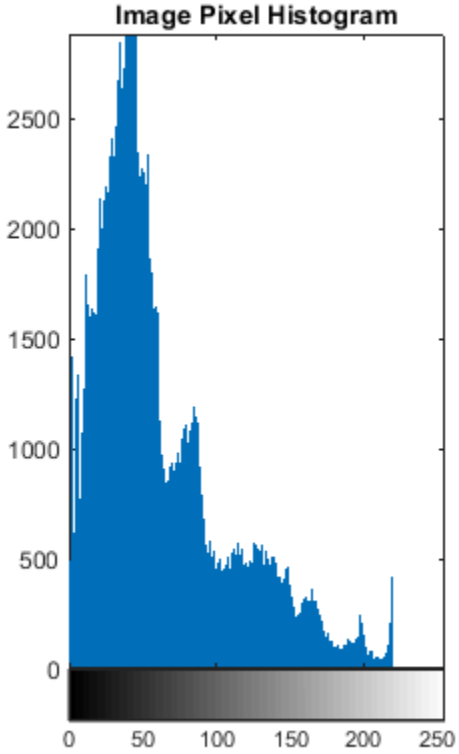



Image3

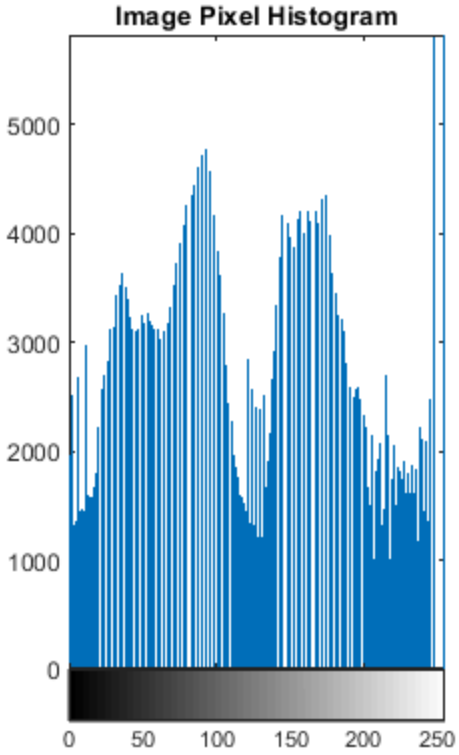


Image Pixel Histogram
5000
4000
3000
2000
1000
0
0
50
100
150
200
250


Image4

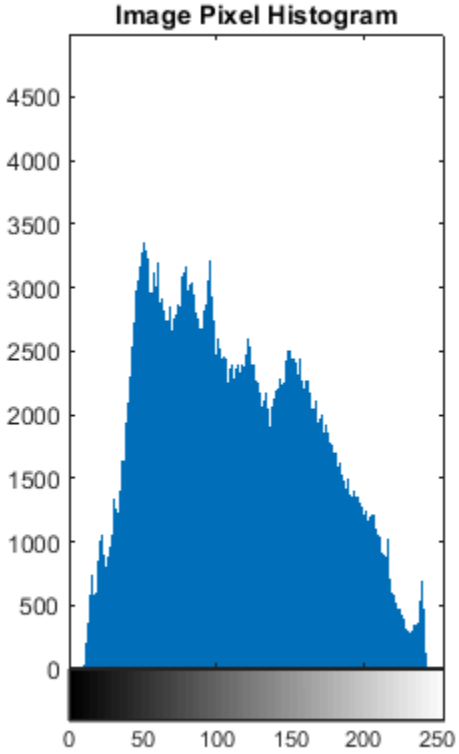
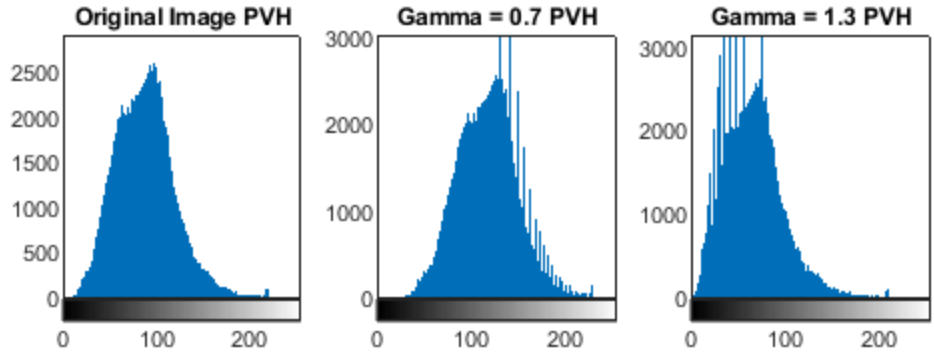


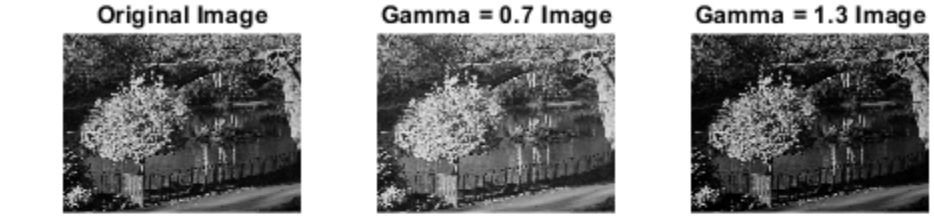
Image Pixel Histogram
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

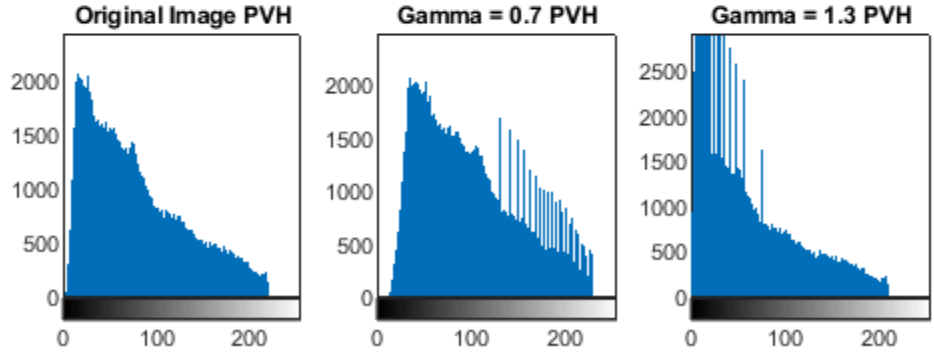
Part 1 - 2nd Task - Gamma Correction

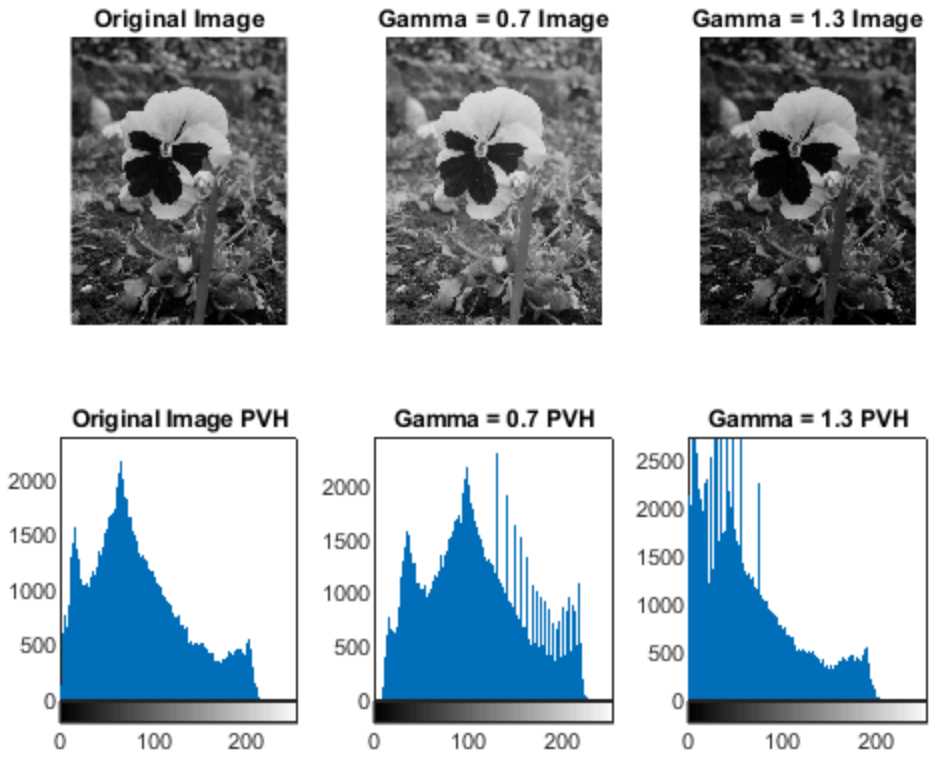
```
P12Imgs = {'unaltIm1.tif','unaltIm2.tif','unaltIm3.tif'};

for k = 1:length(P12Imgs)

    img = imread(P12Imgs{k});
    figure(k);
    subplot(2,3,1);
    imshow(img);
    title('Original Image')

    subplot(2,3,2);
    newimg = gammacorrect(0.7,img);
    newimg = uint8(newimg);
    imshow(newimg)
    title 'Gamma = 0.7 Image'

    subplot(2,3,3);
    newimg = gammacorrect(1.3,img);
    newimg = uint8(newimg);
    imshow(newimg)
    title 'Gamma = 1.3 Image'

    img = imread(P12Imgs{k});
    figure(k);
    subplot(2,3,4);
    imhist(img);
    title('Original Image PVH')

    subplot(2,3,5);
    newimg = gammacorrect(0.7,img);
    newimg = uint8(newimg);
    imhist(newimg)
    title 'Gamma = 0.7 PVH'

    subplot(2,3,6);
    newimg = gammacorrect(1.3,img);
    newimg = uint8(newimg);
    imhist(newimg)
    title 'Gamma = 1.3 PVH'

end
```


Original Image
Gamma = 0.7 Image
Gamma = 1.3 Image


Original Image PVH
Gamma = 0.7 PVH
Gamma = 1.3 PVH
2500
2000
1500
1000
500
0
3000
2000
1000
0
100
200

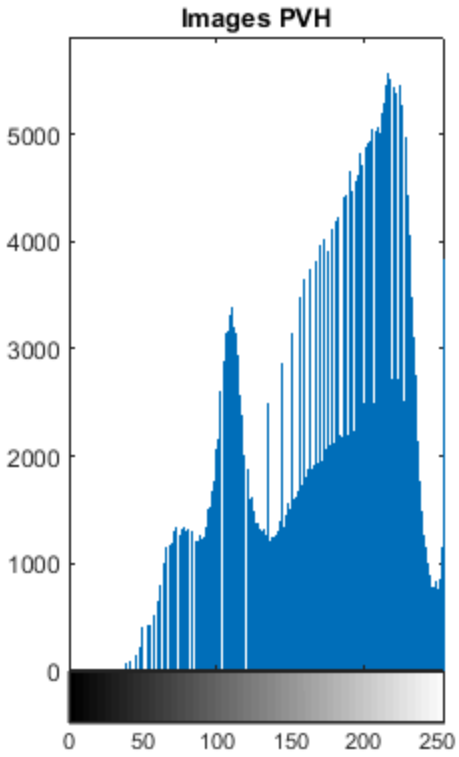

Original Image
Gamma = 0.7 Image
Gamma = 1.3 Image


Original Image PVH
Gamma = 0.7 PVH
Gamma = 1.3 PVH
2000
1500
1000
500
0
2500
100
200



Part 1 - 3rd Task - Contractive or Expansive Mappings

```
P3Img = imread('imageCE5.tif');

figure
subplot(1,2,1);
imshow(P3Img);
title('Image');

subplot(1,2,2);
imhist(P3Img);
title('Images PVH');
```



Part 2 - 1st Task - Detecting Image Resampling and Resizing

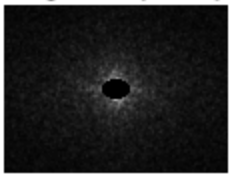
```
P2Imgs = {'resamp1.tif','resamp2.tif','resamp3.tif','resamp4.tif'};
%Load in images

l = 1; % Lamda value
t = 2; % Tau value
s = 1; % Sigma value
%Set values used in function below

for i = 1: length(P2Imgs)
    p_map = kirchnerPmap(P2Imgs{i},l,t,s); %Apply Kirchner P map
 function to obtain p map and frequency p map
end

%Following
type kirchnerPmap.m


function [p] = kirchnerPmap(Image,lamda,tau,sigma)
% This function was created to implement Kirchner's resampling
 detection
% algorithim using a fixed linear prediction filter to approximate
 this
```

```
% relationship


img = double(imread(Image));
[x y] = size(img); % Get image dimensions

alpha_filter = [-0.25 0.5 -0.25
                  0.5 0 0.5
                -0.25 0.5 -0.25];

newimg = filter2(alpha_filter, img); % Apply the alpha filter
Error = img - newimg; % Calculate the error
p = lamda*exp(-Error.^tau/sigma); % Obtain the p-map

figure;
subplot(3,1,1);
imshow(uint8(img));%show the original img
title(string(Image));

subplot(3,1,2);
imagesc(p) % display the calculated p-map
title('Images P Map');

%The axis of plots must be normalized to allow for comparison, as done
%below

axis equal
axis off;
xlim([1 y]);
ylim([1 x]);

subplot(3,1,3);
showFreqPmap(p);
title('Images Freq P map');

axis equal
axis off;
xlim([1 y]);
ylim([1 x]);
end
```

resamp1.tif

Images P Map

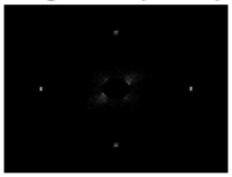
Images Freq P map



resamp2.tif

Images P Map

Images Freq P map



resamp3.tif

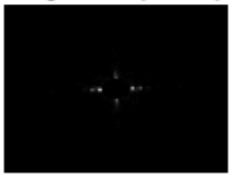
Images P Map

Images Freq P map

resamp4.tif

Images P Map

Images Freq P map



Published with MATLAB® R2019b