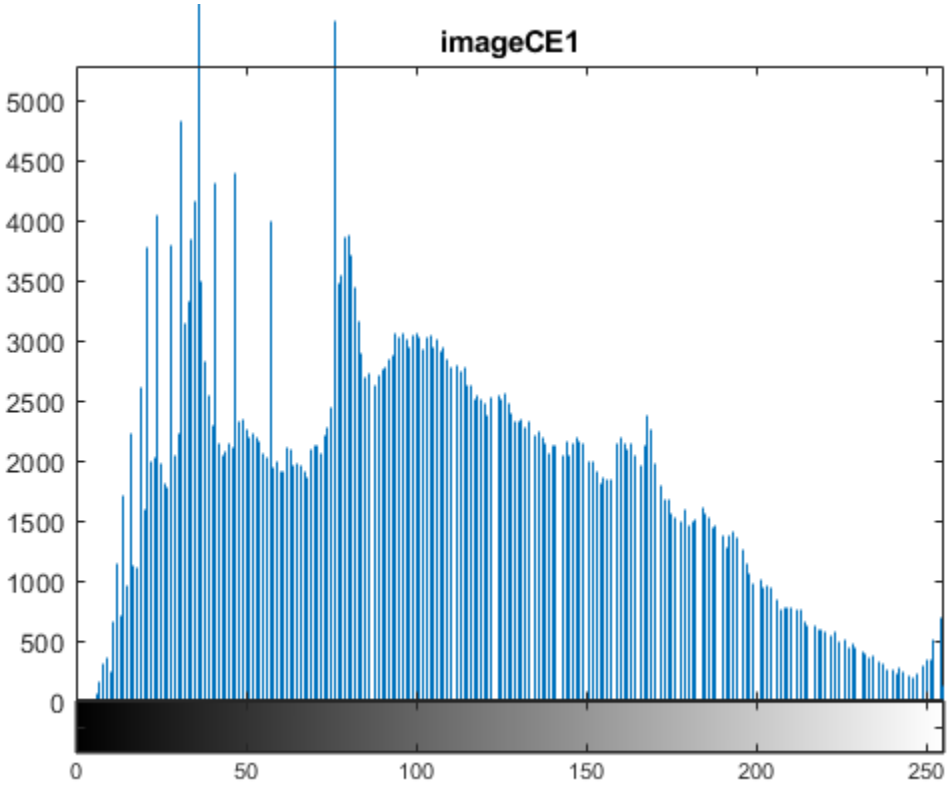
Kleanthis E Karavangelas, Basem Saleh

Assignment 5, ECES 435

Part 1: Detecting Image Contrast Enhancement

imageCE1.tif

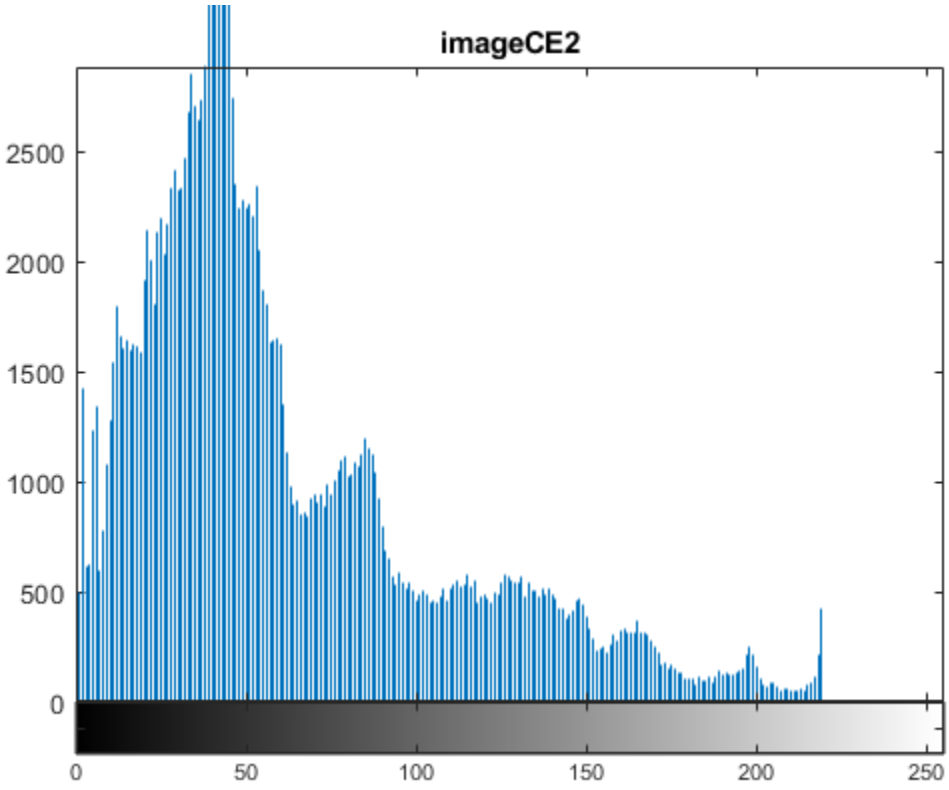
```
ce1 = imread('imageCE1.tif');
figure(1);
imhist(ce1);
title("imageCE1");
```



imageCE2.tif

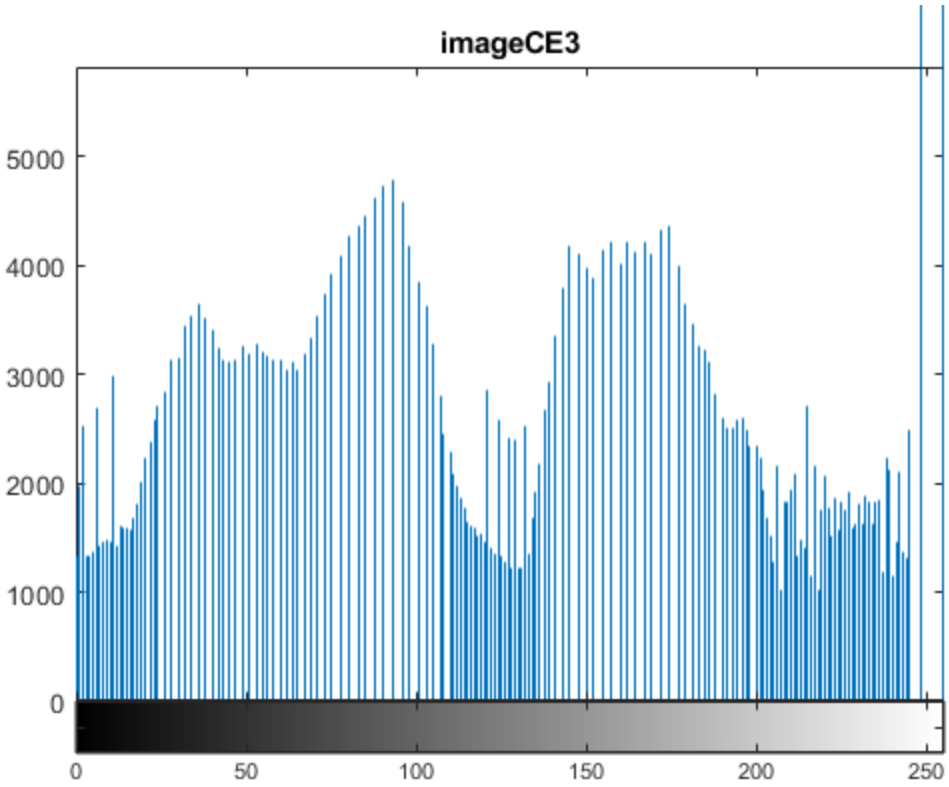
```
ce2 = imread('imageCE2.tif');
figure(2);
```

```
imhist(ce2);
title("imageCE2");
```



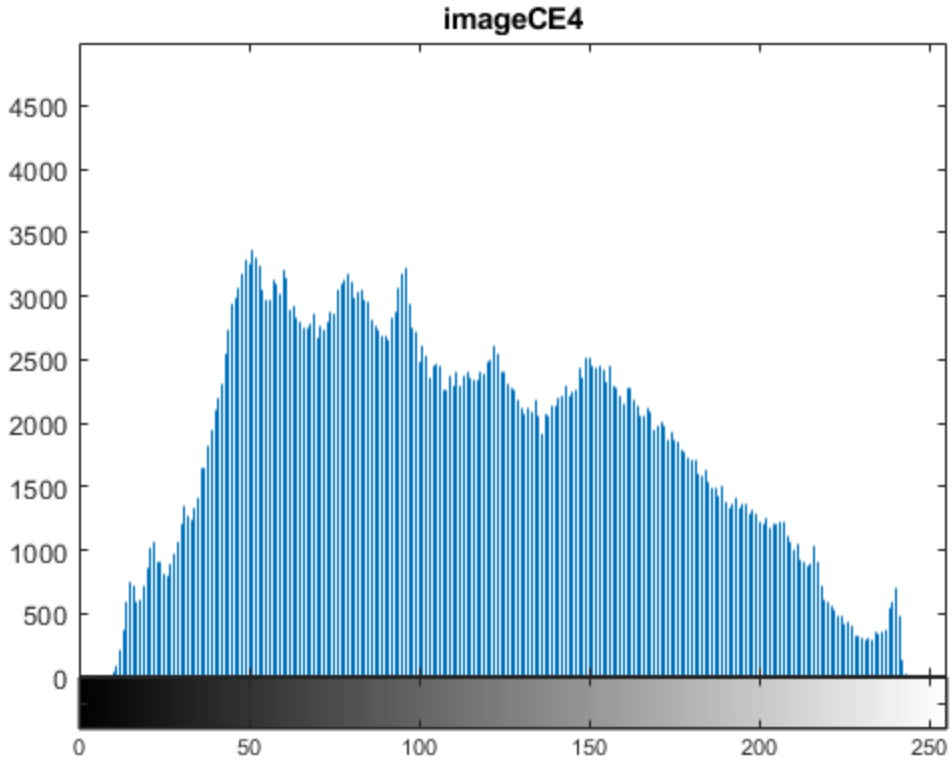
imageCE3.tif

```
ce3 = imread('imageCE3.tif');
figure(3);
imhist(ce3);
title("imageCE3");
```



imageCE4.tif

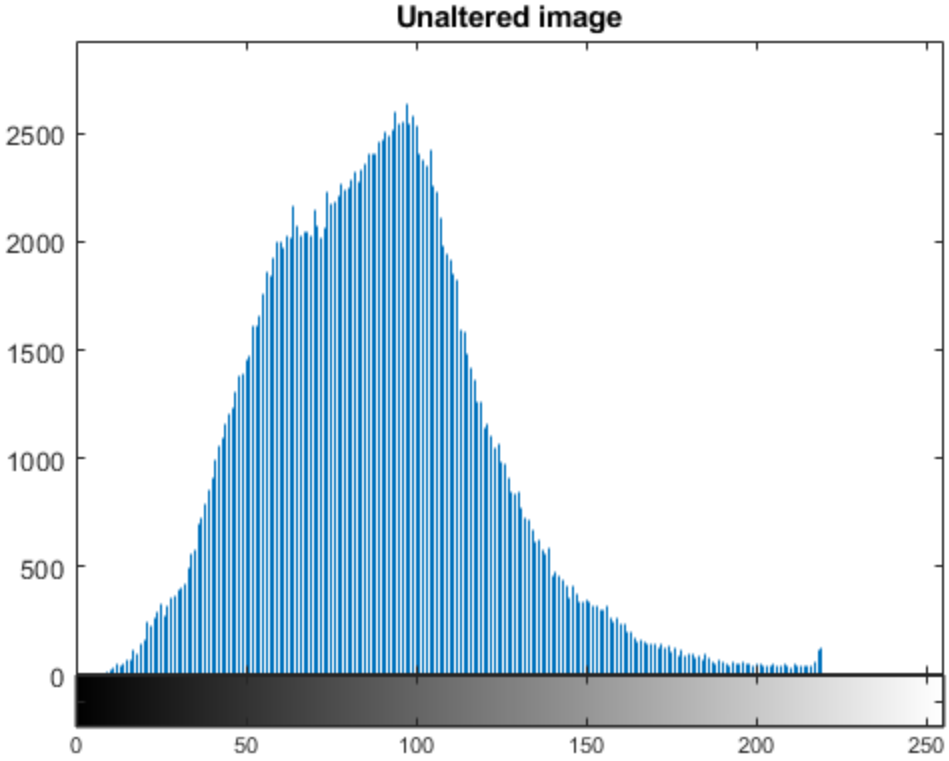
```
ce4 = imread('imageCE4.tif');
figure(4);
imhist(ce4);
title("imageCE4");
```

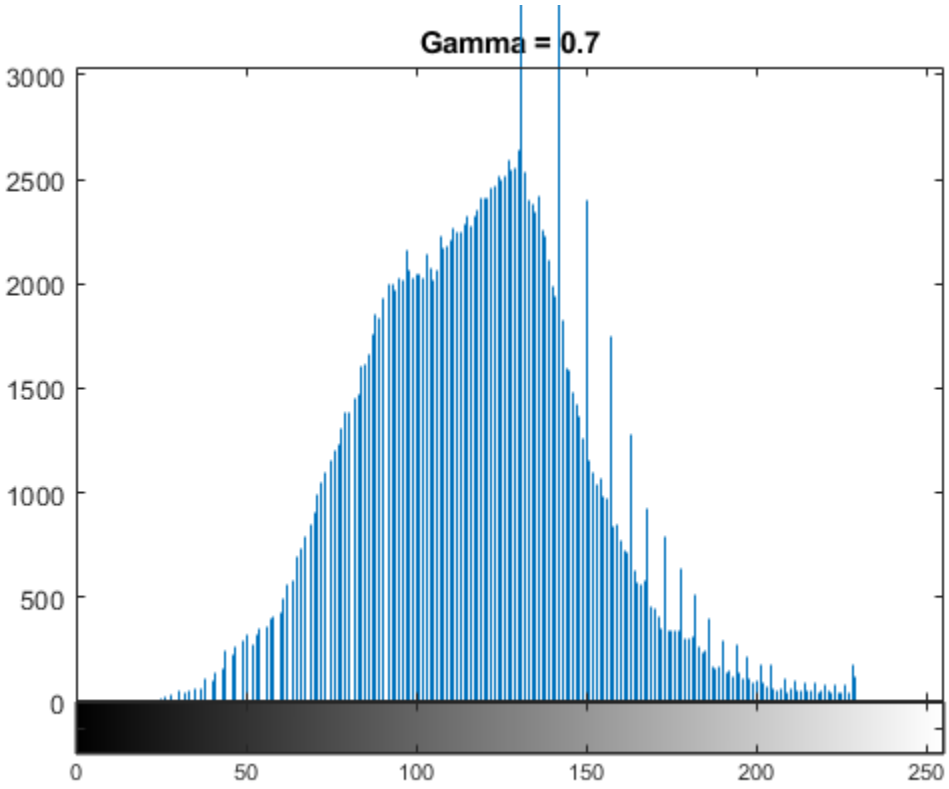


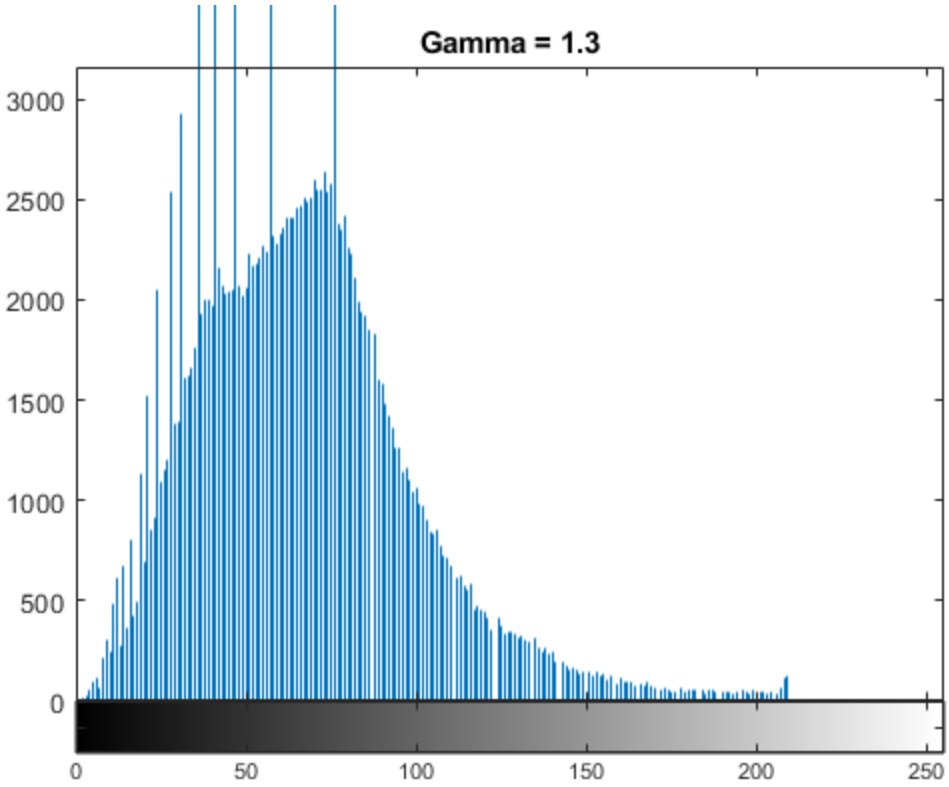
Based on the histograms generated, it seems that images 1 and 3 have been contrast enhanced. This is because you can observe gaps in the histogram, which are an indicator of contrast enhancement.

unaltIm1.tif

```
un1 = imread('unaltIm1.tif');
figure(5);
imhist(un1);
title('Unaltered image');

g07 = GammaCorrection('unaltIm1.tif', 0.7);
figure(6);
imhist(g07);
title('Gamma = 0.7');

g13 = GammaCorrection('unaltIm1.tif', 1.3);
figure(7);
imhist(g13);
title('Gamma = 1.3');
```

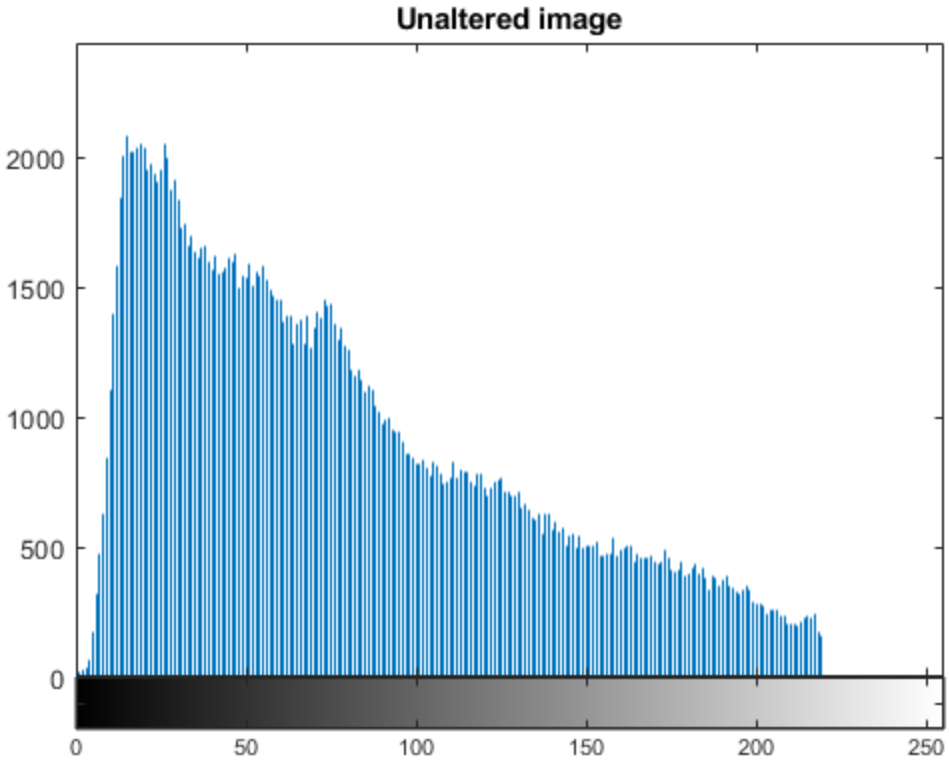

Unaltered image
2500
2000
1500
1000
500
0
0
50
100
150
200
250

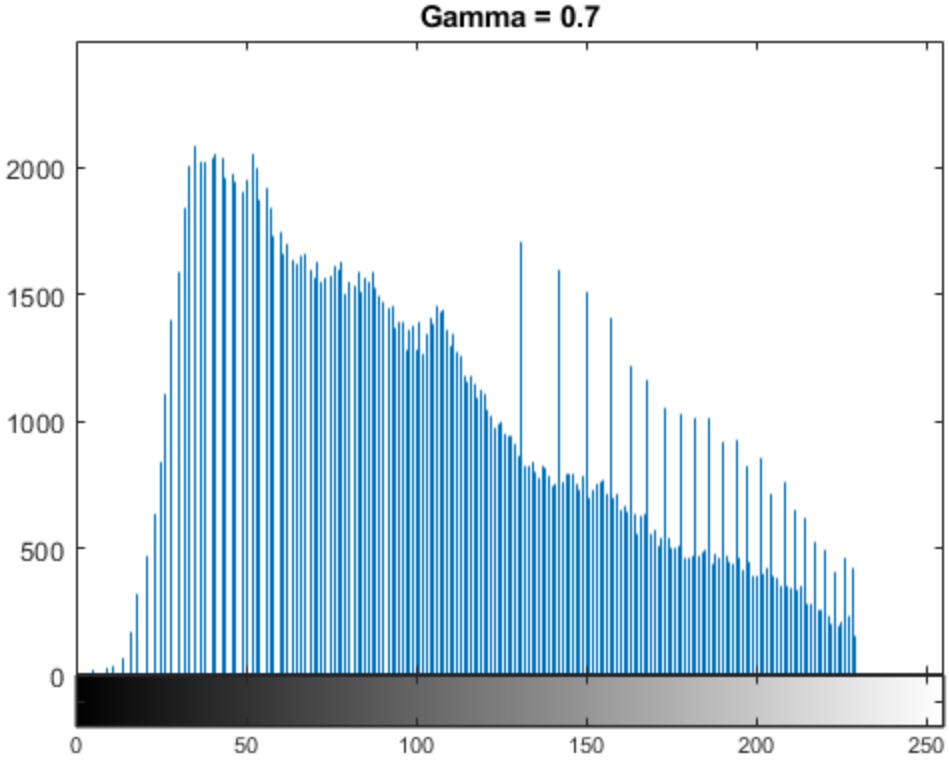

Gamma = 0.7
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

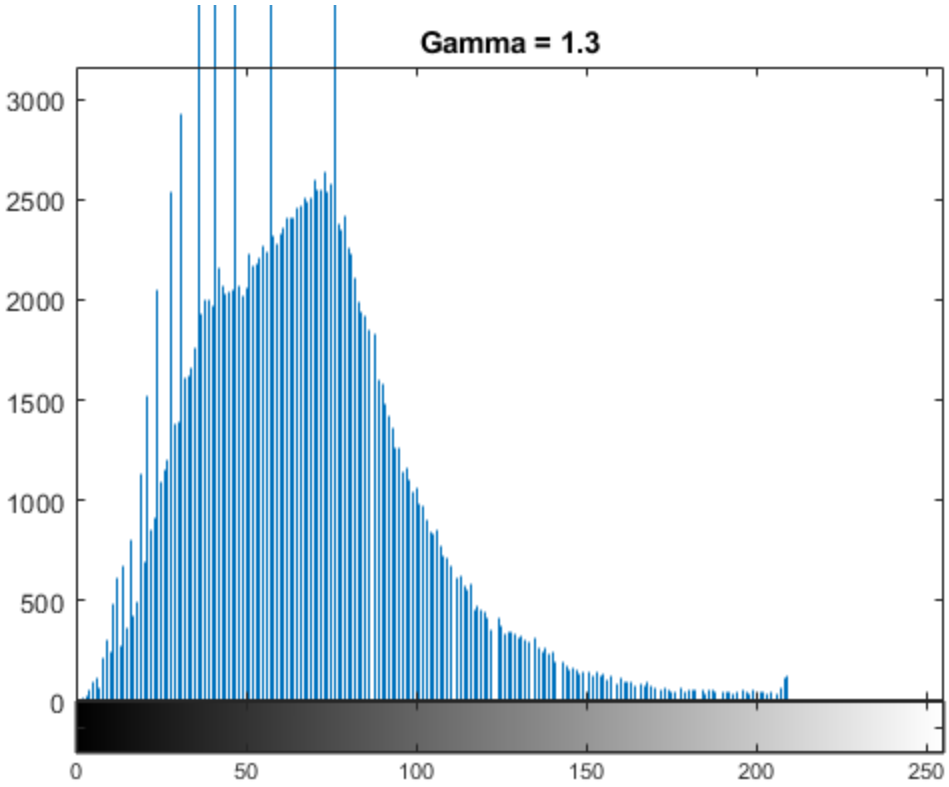


unaltIm2.tif

```
un2 = imread('unaltIm2.tif');
figure(8);
imhist(un2);
title('Unaltered image');

g207 = GammaCorrection('unaltIm2.tif', 0.7);
figure(9);
imhist(g207);
title('Gamma = 0.7');

g213 = GammaCorrection('unaltIm2.tif', 1.3);
figure(10);
imhist(g213);
title('Gamma = 1.3');
```

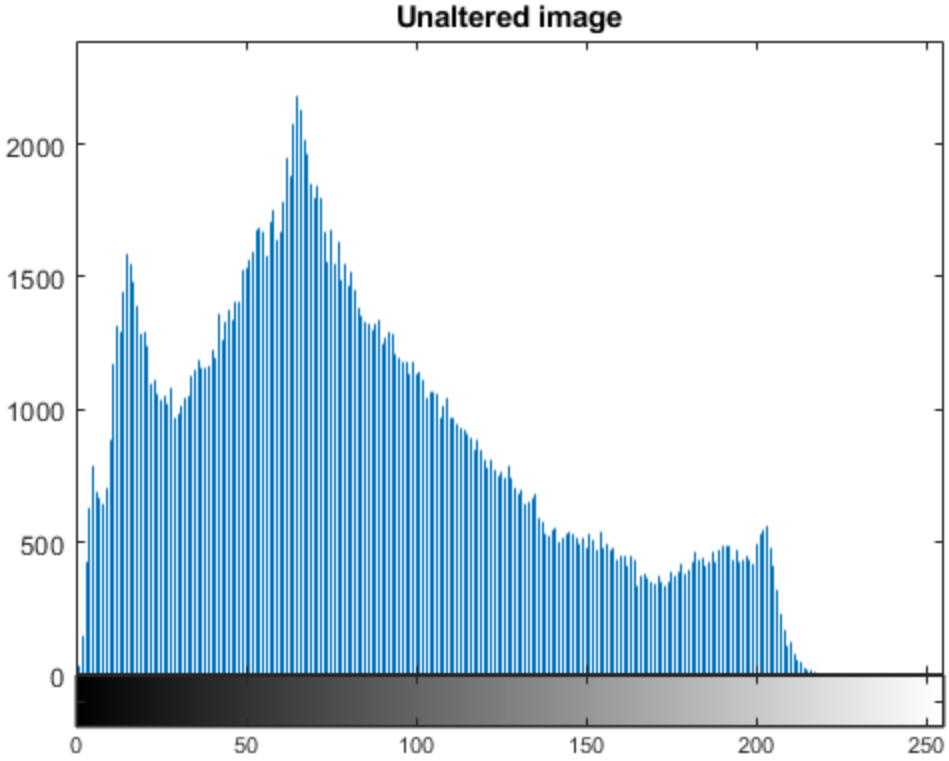

Unaltered image
2000
1500
1000
500
0
0
50
100
150
200
250

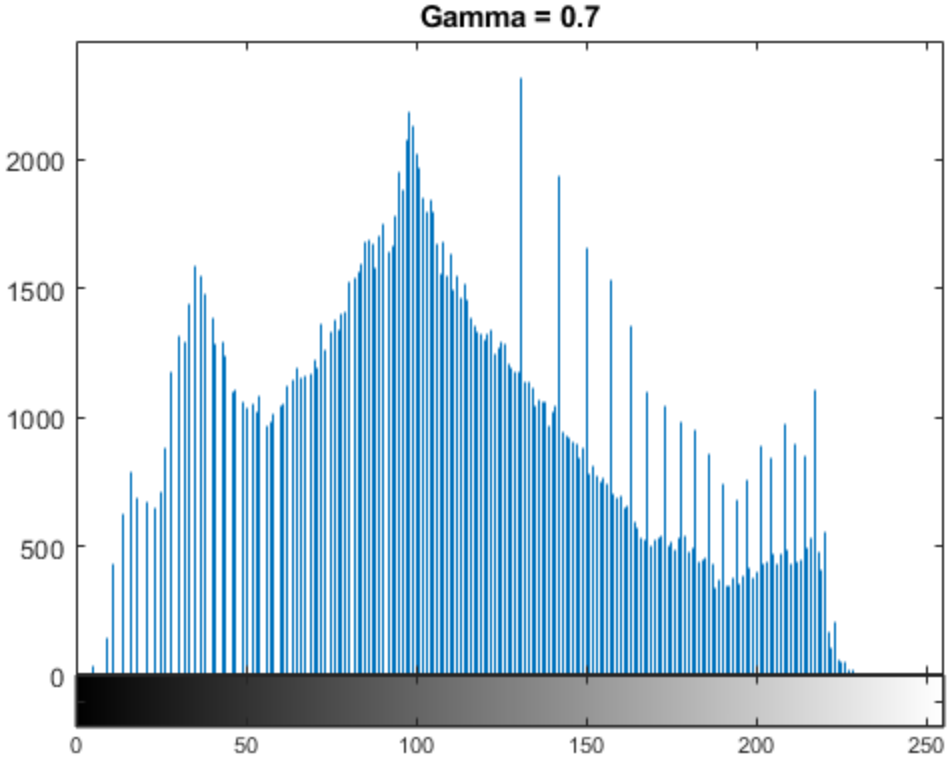

Gamma = 0.7
2000
1500
1000
500
0
0
50
100
150
200
250

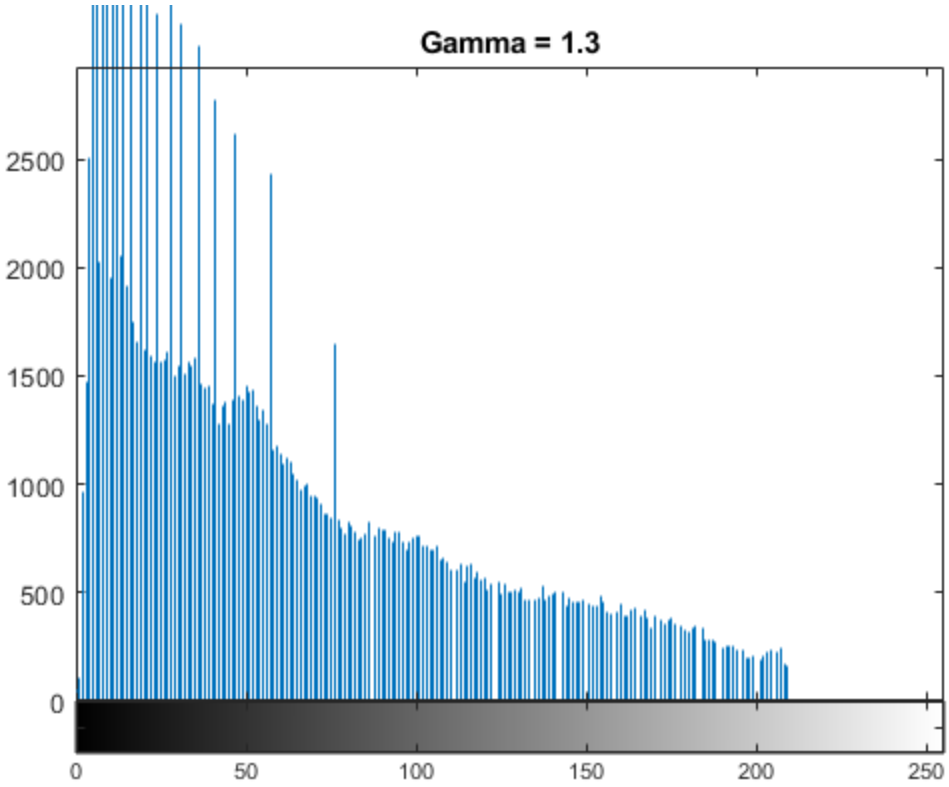


unaltIm3.tif

```
un3 = imread('unaltIm3.tif');
figure(11);
imhist(un3);
title('Unaltered image');

g307 = GammaCorrection('unaltIm3.tif', 0.7);
figure(12);
imhist(g307);
title('Gamma = 0.7');

g313 = GammaCorrection('unaltIm3.tif', 1.3);
figure(13);
imhist(g313);
title('Gamma = 1.3');
```


Unaltered image
2000
1500
1000
500
0
0
50
100
150
200
250

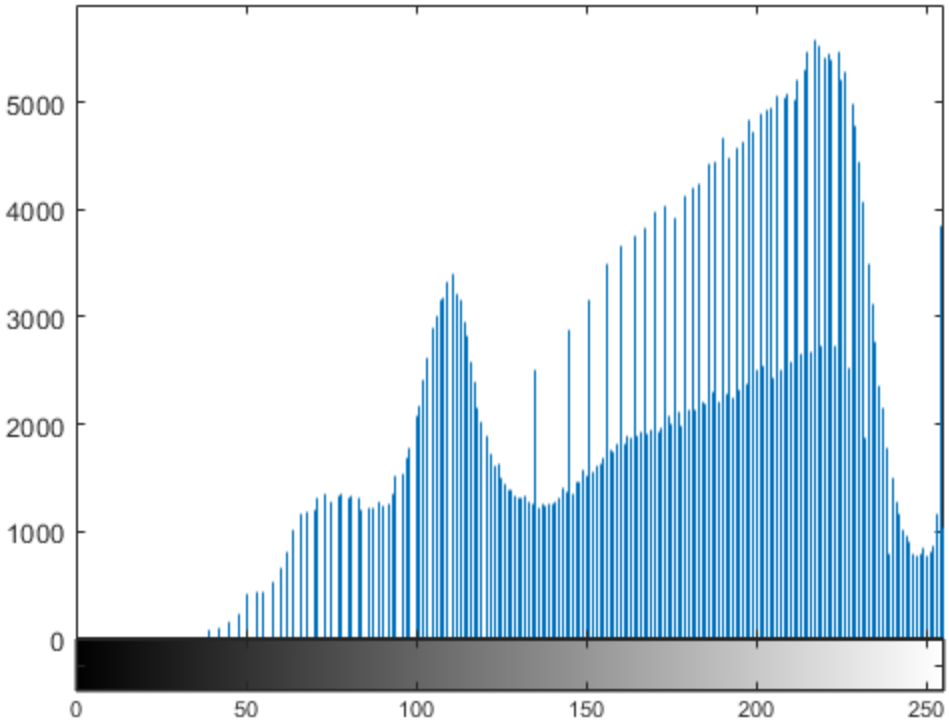

Gamma = 0.7
2000
1500
1000
500
0
0
50
100
150
200
250



For gamma = 0.7 and therefore less than 1, we can see that the gaps appear on the left side of the histogram, and we observe amplitude spikes on the right side. For gamma = 1.3 and therefore more than 1, we can see that the gaps appear on the right side of the histogram, and we observe amplitude spikes on the left side of the histogram.

imageCE5.tif

```
ce5 = imread('imageCE5.tif');
figure(14);
imhist(ce5);
```

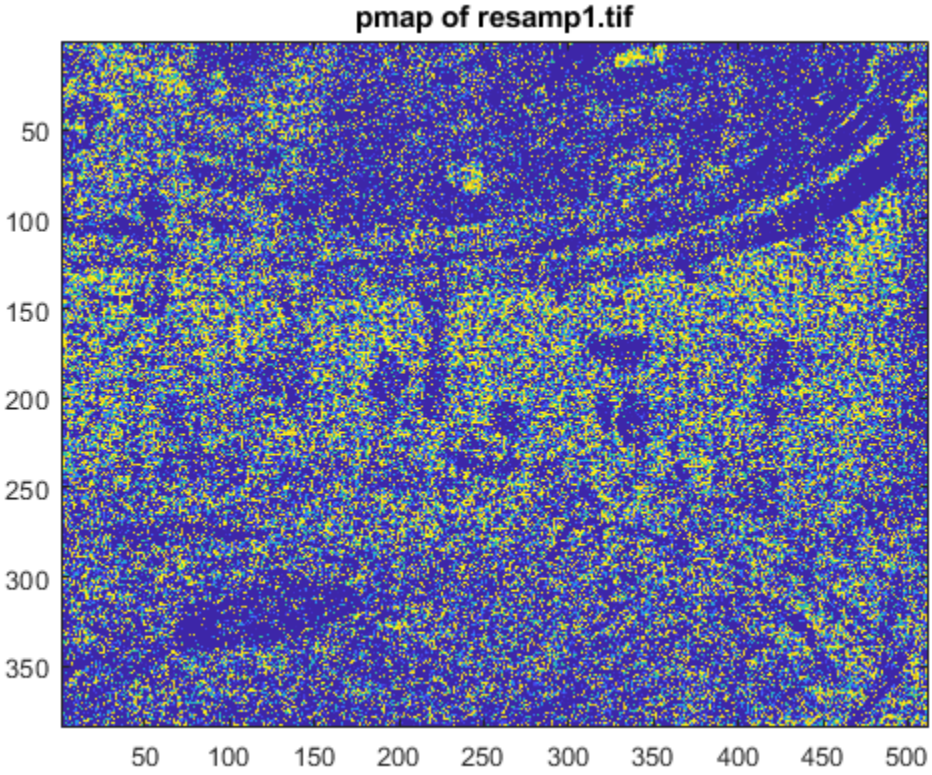


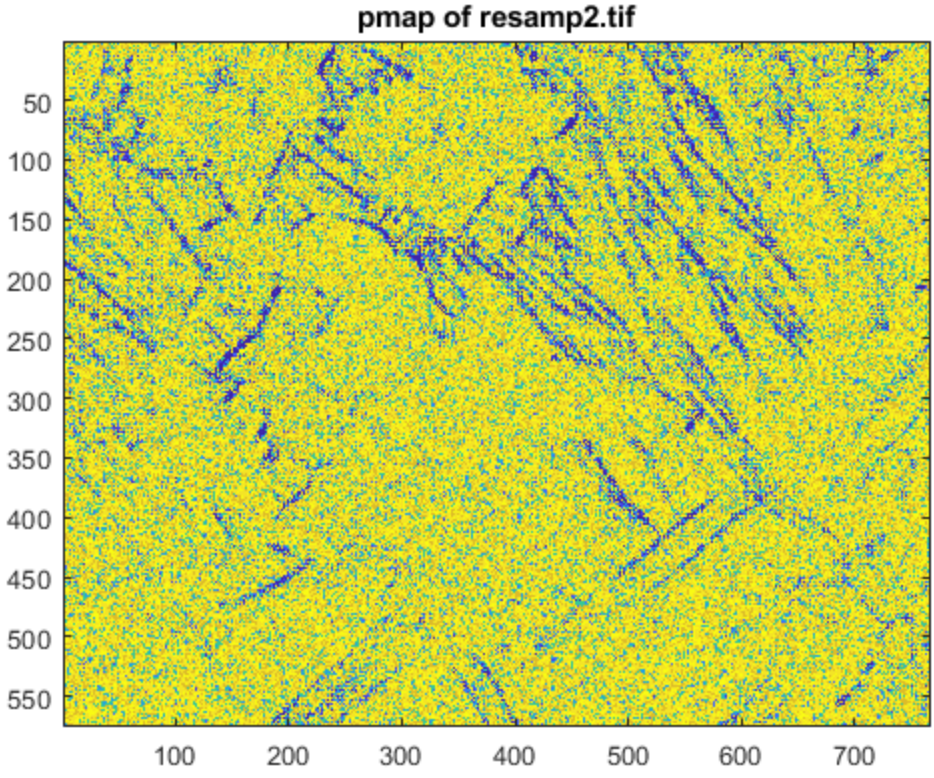
Based on the histogram, gamma is less than one (gamma<1).

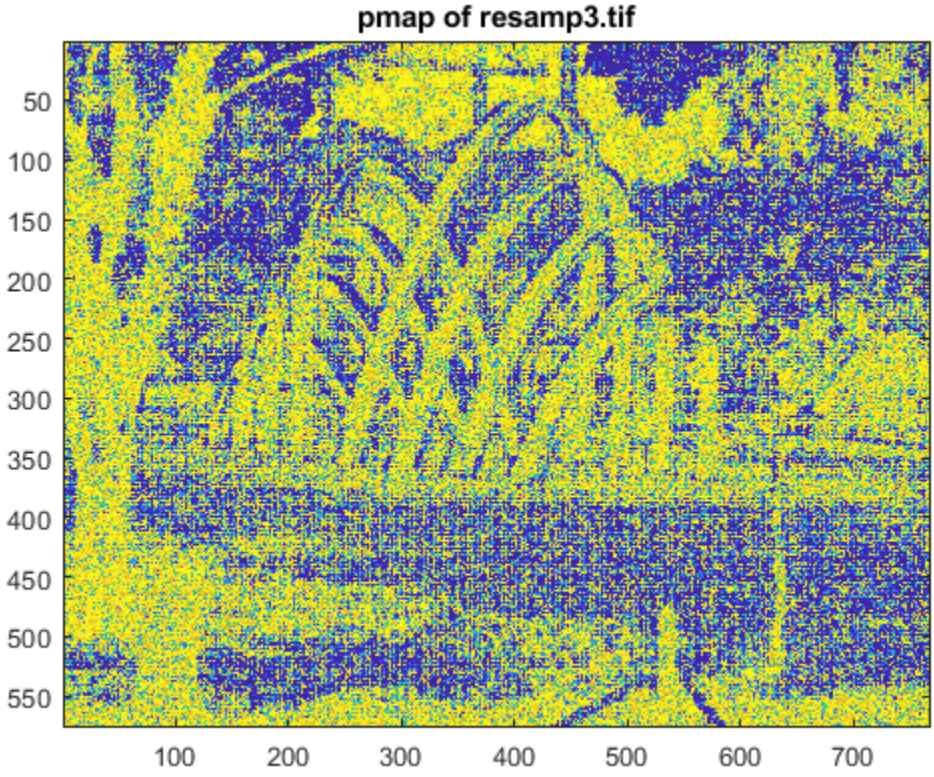
Part 2: Detecting Image Resampling and Resizing

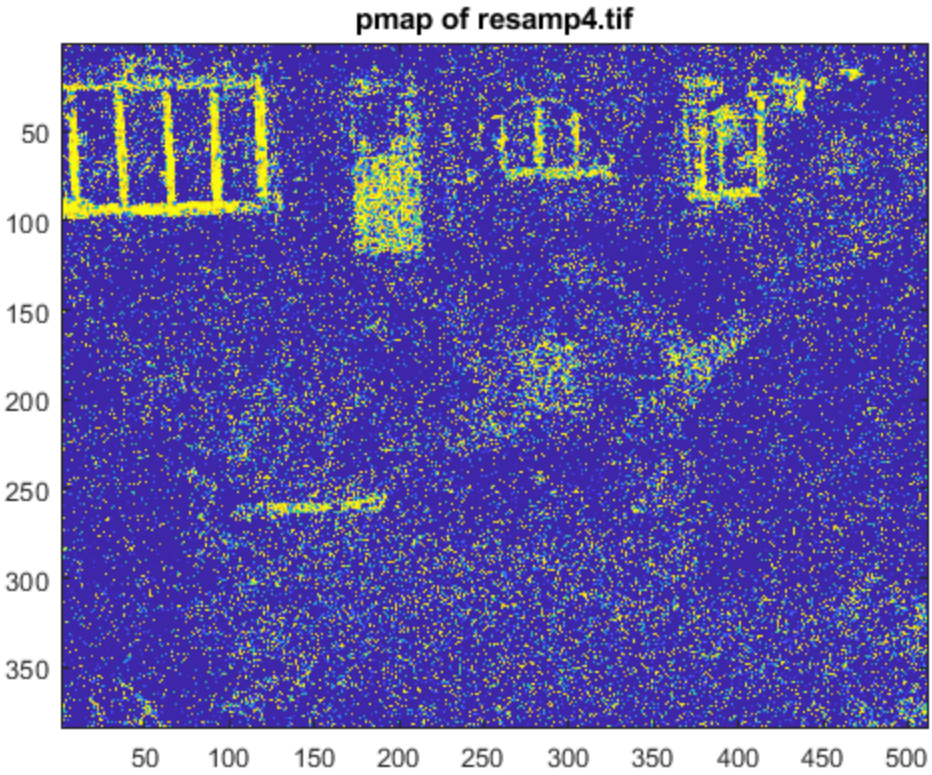
```
% resamp1.tif
pmap1 = Kirchner('resamp1.tif');
figure(15);
imagesc(pmap1);
title('pmap of resamp1.tif');

% resamp2.tif
pmap2 = Kirchner('resamp2.tif');
figure(16);
imagesc(pmap2);
title('pmap of resamp2.tif');

% resamp3.tif
pmap3 = Kirchner('resamp3.tif');
figure(17);
imagesc(pmap3);
title('pmap of resamp3.tif');

% resamp4.tif
pmap4 = Kirchner('resamp4.tif');
figure(18);
```

```
imagesc(pmap4);
title('pmap of resamp4.tif')
```




pmap of resamp2.tif
50
100
150
200
250
300
350
400
450
500
550
100
200
300
400
500
600
700

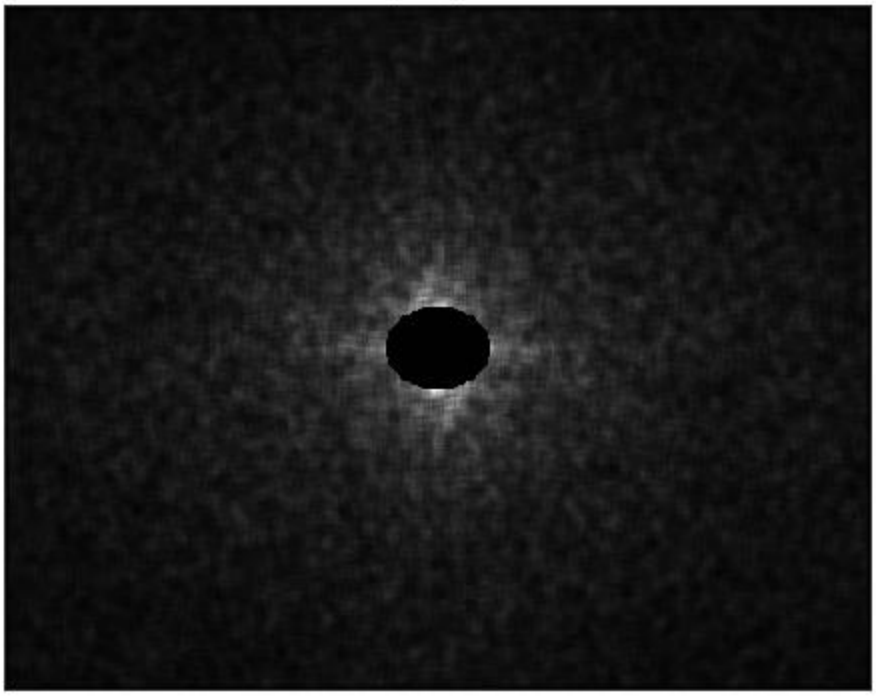

pmap of resamp3.tif
50
100
150
200
250
300
350
400
450
500
550
100
200
300
400
500
600
700



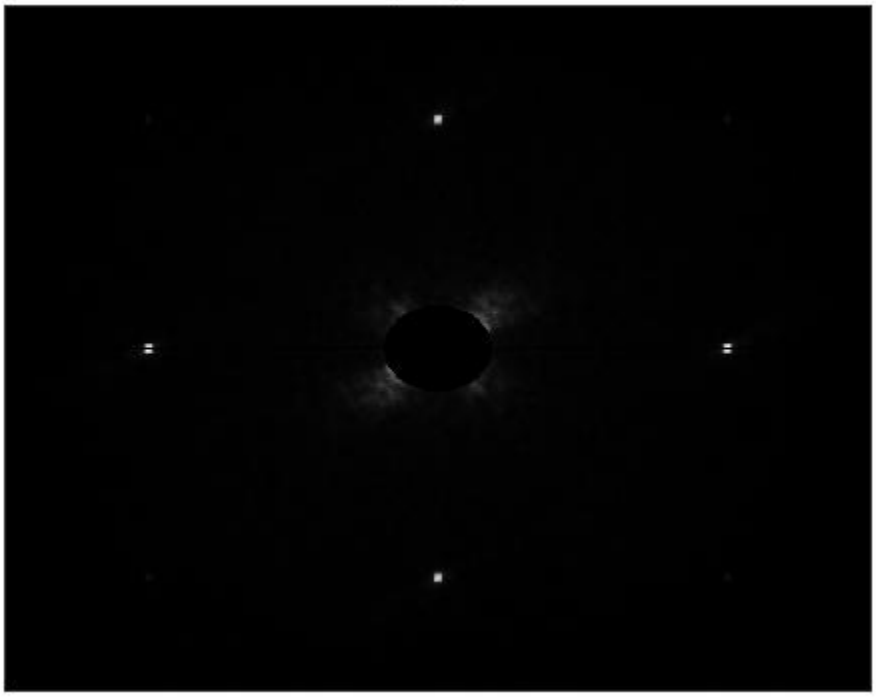
pmaps

```
figure(19);
showFreqPmap(pmap1);
title('pmap1');
figure(20);
showFreqPmap(pmap2);
title('pmap2');
figure(21);
showFreqPmap(pmap3);
title('pmap3');
figure(22);
showFreqPmap(pmap4);
title('pmap4');
```

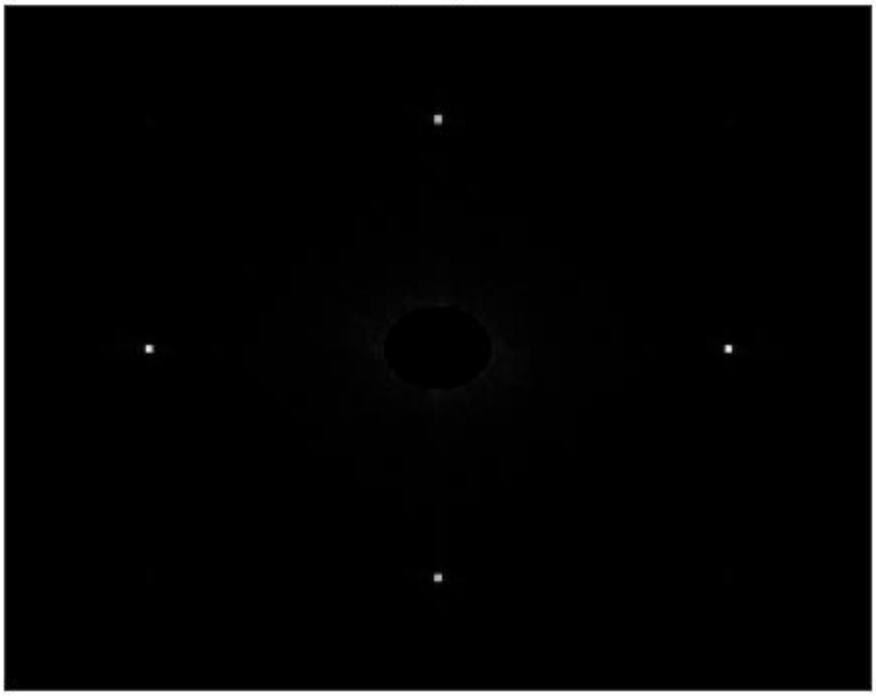
pmap1



pmap2



pmap3



pmap4

Based on the magnitude of the Fourier transform's generated using the pmaps of the previous images, we can tell that images 2 and 3 have been resampled. The fingerprints indicating this are the periodical artifacts that were transformed as peaks in the frequency domain, or otherwise the white dots you are able to see in the images 2 and 3.

Kirchner's algorithm

```
function pmap = Kirchner(Img)

    % first we read the image
    Im = imread(Img);

    % then convert it from uint8 to double so we can modify it
    Im = double(Im);

    % create the linear prediction filter
    Lpf = [-0.25 0.5 -0.25;
        0.5 0 0.5;
        -0.25 0.5 -0.25];

    Imp = filter2(Lpf, Im);

    err = Im - Imp;

    pmap = 1*exp((-err.^2)/1);

end
```

Gamma Correction function

```
function gc = GammaCorrection(Img, Gamma)

    % first, we read the image passed into the function
    x = imread(Img);

    % we then convert the data from unit8 to double, so we can apply the
    % mapping
    x = double(x);

    % here we modify each pixel value, with the following expression
    gc = 255*(x/255).^Gamma;

    % convert back to unit8 in order to be able to show the image
    gc = uint8(gc);

end
```

Published with MATLAB® R2019b