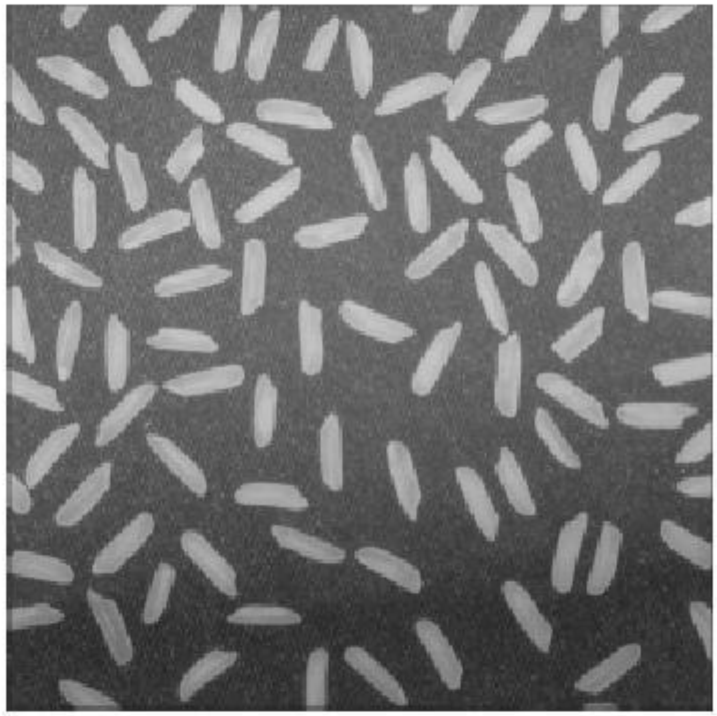
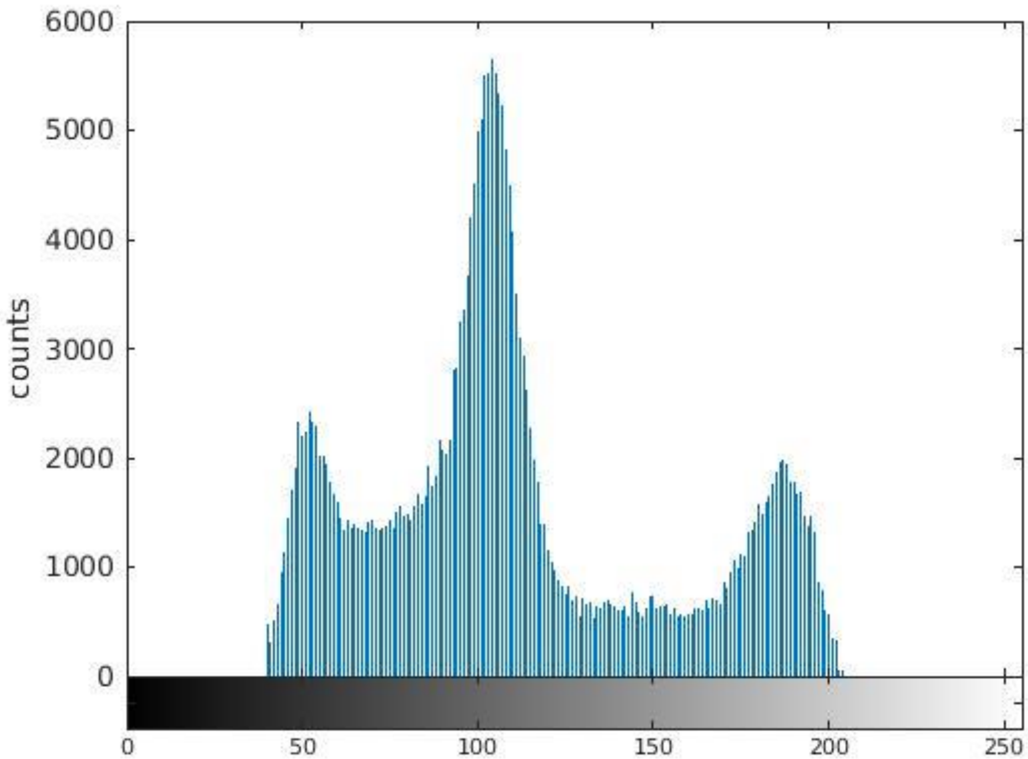
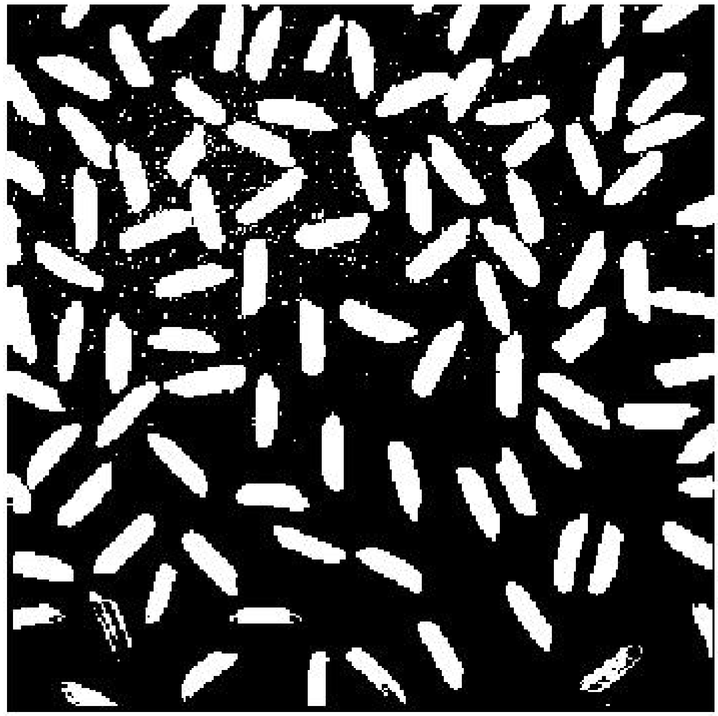
1) Display of the image and the histogram graph of the image as below:



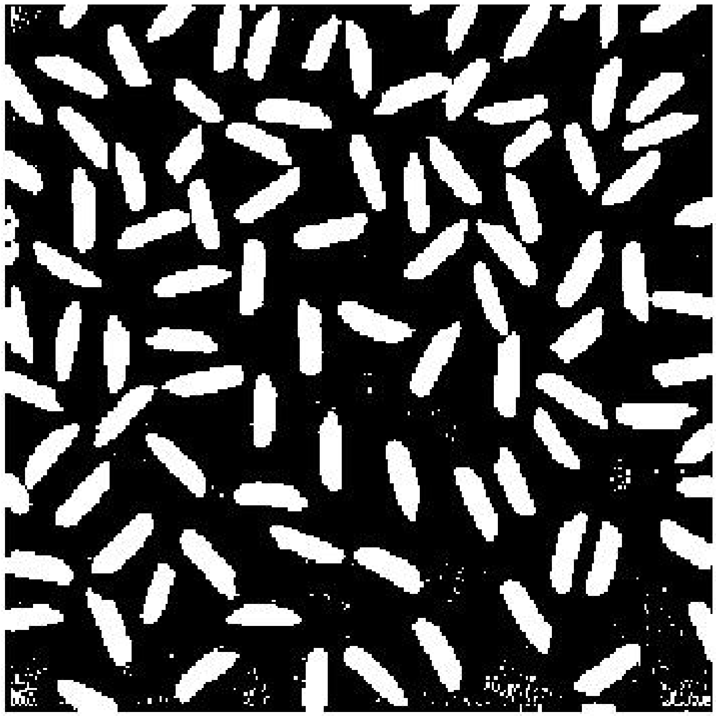


a) In binarizing the instensity values of the image, if we choose a global trheshold it should be between 125 and 118. If we choose a threshold greater than 125 , the rices at the lower half of the image get to lose their brightness and so their visibility. If we choose a threshold smaller than 118, some background noises brighten at the upper half of the image and damage the clearence in the visibility of rices on this side of the image. Since 122 as a threshold which minimimizes the background noises at the upper half of the image and maximizes the shape and brightness of the rices at the lower half of the image, I chose this number as the threshold value in my enhancement work. This value also could be justified with the help of the histogram. As it has shown, the intensity level around 120 is the kind of threshold which specifies the end of the majority of pixels in the dark side and start of the majority of pixels in the bright side. With the help of some manual adjustment in treshold, it could be decided on 122 as a treshold.



a) with a global threshold (122/255)

b) Other way to ehance the image by binarizing could be applying adaptive threshold strategy to this image since the lower side of the image has a shadow on the rices and it causes important differences in general intensity level of upper and lower sides of the image. These differences in tonality between two halves of the image requires us to use local thresholds to maximize the improvement in enhancement quality of the image. In matlab code, if we choose brighter foreground settings with the imbinarize function’s adaptive option, we will obtain the best version of display for rices.



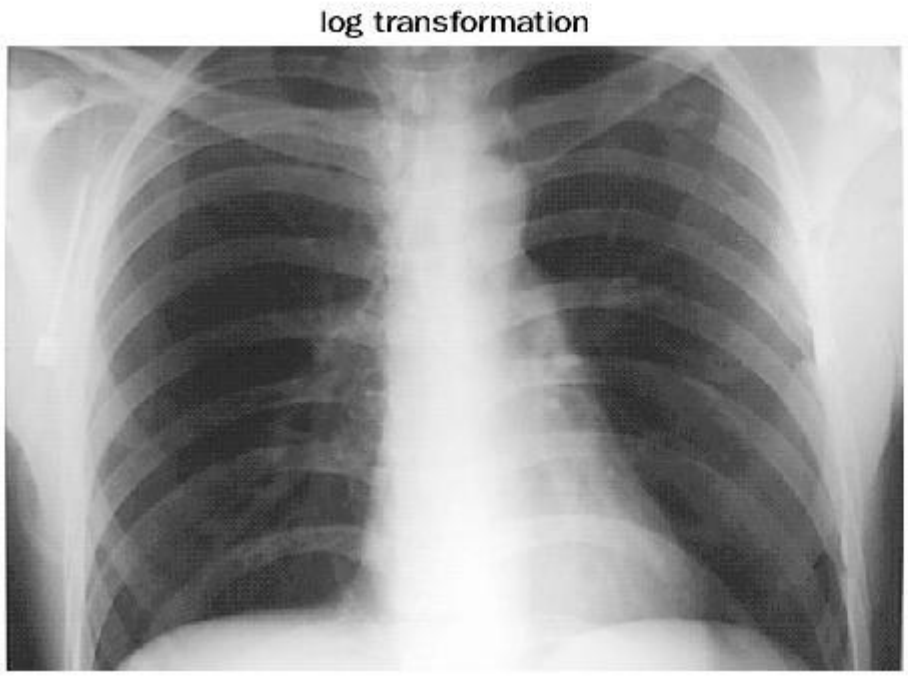
b)with adaptive threshold

2) At first glance, it seems like there are some important details in the dark side of the image in which part the bones are supposed to be clear for vision. But they are too dark to be recognized.

   a) Since the details that we care about are hidden in the dark side and we need to brighten them to have them explicit, some nonlinear functions such as logarithmic transformation or nth power functions could be useful to enhance the image. Their common feature of these functions is they are brightening each intensity especially darker ones at a steep rate. With the help of them, we can make the details in the dark part more visible.

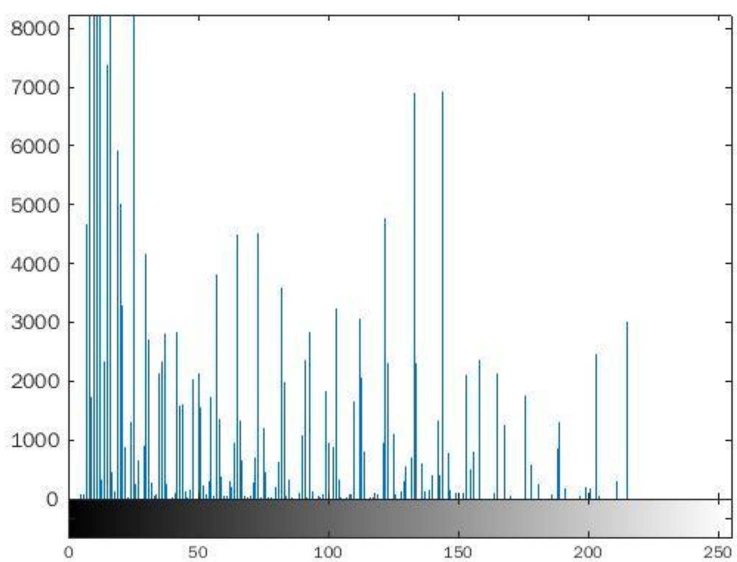
Application of Logarithmic Function as a transformation function to enhance the image:

```
img2=imread('chestXray-dark.tif');
imshow(img2)
imhist(img2)
%%log_correction%%
g=log(1+double(img2))
g=(g-min(g(:)))/(max(g(:))-min(g(:)))*256
g=uint8(g);
figure; imshow(g), title('log transformation')
```



b) Another application could be histogram equalization. When we evaluate the histogram of the original image, we see that density of the distribution of intensity levels is much higher between the bins 0-50.

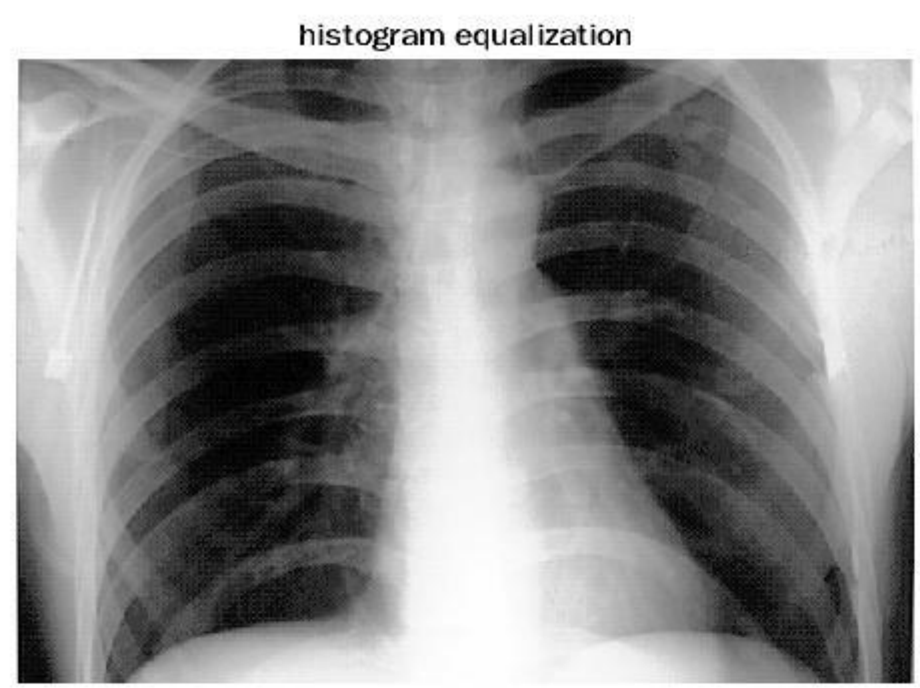
Histogram of gray levels of **original** image:



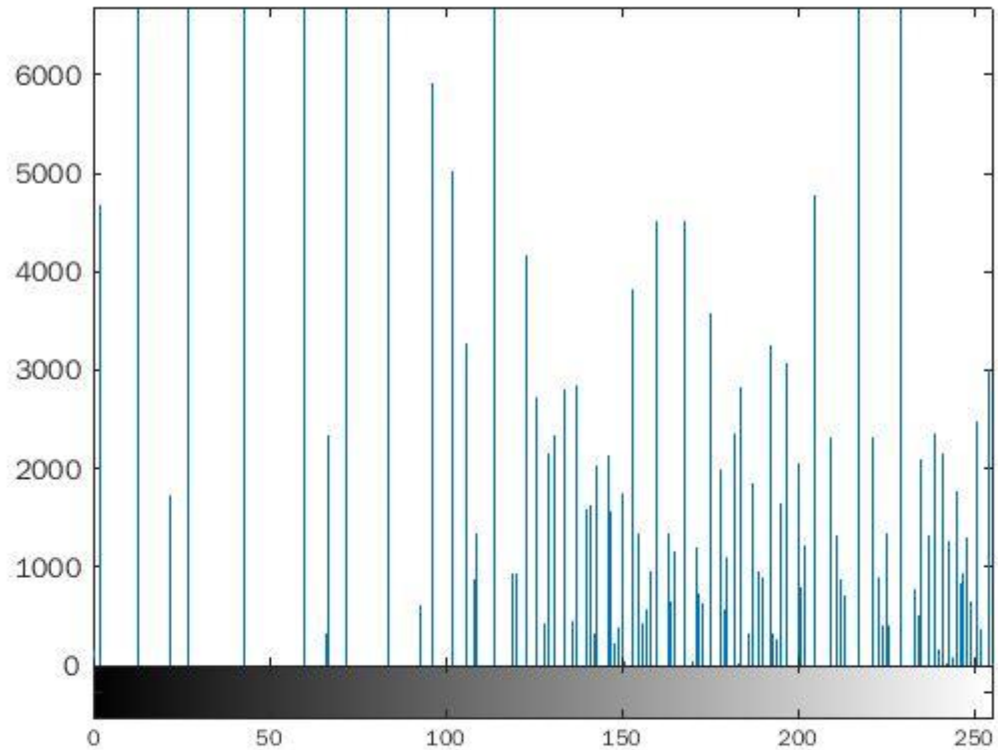
When we consider that details, we need to know more about are hidden in the dark part and the information that histogram equalization would stretch out especially the dense part of the histogram, we can conclude that histogram equalization could also help us brighten the details of bones.

Application of Histogram Equalization transformation in order to enhance the image:

```
h=histeq(img2,256)
figure; imshow(h), title('histogram equalization')
```



Histogram of the intensity levels of the image after histogram equalization is applied:



As it has shown, the dark side of the image that were dense between bins 0 and 50 are stretched out.

Full Matlab Script:

```
%%Q1%%
img = imread('rice.tif');
imshow(img)
[R, C]=size(img),class(img)
imhist(img)
xlabel('intensity');
ylabel('counts');
xlim([0,255]);
ylim([0,6000]);


%%ALTERNATIVE ADAPTIVE%%
imshow(imbinarize(img,125/255))
imshow(imbinarize(img,"adaptive","ForegroundPolarity","bright"))
```

```
%%Q2%%
img2=imread('chestXray-dark.tif');
imshow(img2)
imhist(img2)
%%log_correction%%
g=log(1+double(img2))
g=(g-min(g(:)))/(max(g(:))-min(g(:)))*256
g=uint8(g);
figure; imshow(g), title('log transformation')
%%histogram_equalization%%
h=histeq(img2,256)
figure; imshow(h), title('histogram equalization')
imhist(h)
```