Image Processing
lab 1

Oscar Hansson 50%   Risto Vehviläinen 50%

December 3, 2018

Exercise 1 – Reducing the number of intensity levels

a. First part of Exercise 1 asked us to reduce the number of intensity levels from 256 to 2. A gray-scale image with 256 levels of intensity is an NxN matrix with values ranging from 0 to 255. For the 2-level case we want the matrix to only contain the number 0 (black) or 255 (white). Taking the 2-level intensity as an example we first need to categorize the different parts of the image into either a black or white pixel. First we reduce the numbers in the matrix by choosing a denominator as 256/numoflevels, which in the case of 2-level intensity becomes $256/2 = 128$. By doing this we can reduce the numbers of the matrix to either be above or below 1. For example the number 255 divided by 128 is 1,9921875 and the number 120 divided by 128 is 0,9375. We are still not quite there since we need to have only two possible pixel values (0 and 255). By using the built-in 'floor' function, which rounds the number down to the closest integer we can effectively make our matrix to only contain zeros and ones. Finally we divide the reduced image by the maximum value in that matrix which in this case will be 1 and then we scale it back up by multiplying with 255. This results in a matrix filled with only 255 and 0.

```matlab
function reducedimage = IPreduce(image, numoflevels)
    denominator = 256 / numoflevels;
    reducedimage = double(floor(double(image)./(denominator)));
    reducedimage = uint8(reducedimage ./ max(reducedimage(:)) .* 255)
        ;
end
```

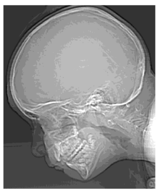
b.

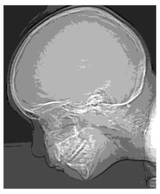
256 levels

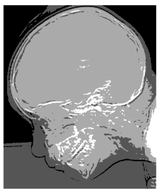
128 levels

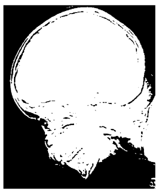
64 levels

32 levels


16 levels


8 levels


4 levels


2 levels

Exercise 2 – Enhancement

a. Contrast stretching normalizes the image and stretches the lower and upper bound of the intensity values to span a broader range of intensity values and thus increasing the contrast of the image. The upper and lower bounds in this case are the minimum and maximum intensity values of the image. The formula for contrast stretching is shown below.[1]

$$\mathrm{f}_{out}(x,y) = 2^N - 1/(M-m) * (f_{in}(x,y) - m)$$

Since we are only working with 8 bit images we use a fixed 2 to the power of 8 in the function for contrast stretching. This could be improved by checking the image beforehand if it is either a 8-bit or 16-bit image for example and adjust the formula accordingly.

```
function stretchedimage = IPcontraststretch(image)
    M = max(image(:));
    m = min(image(:));
    stretchedimage = ((2^8 - 1) / (M-m)) * (image - m);
end
```

[1]Digital Image Processing 3rd ed. - R. Gonzalez, R. Woods, pp. 137

b. Figure depicting the original and the contrast stretched image side-by-side as well as the corresponding histograms for the different intensity levels of the images.

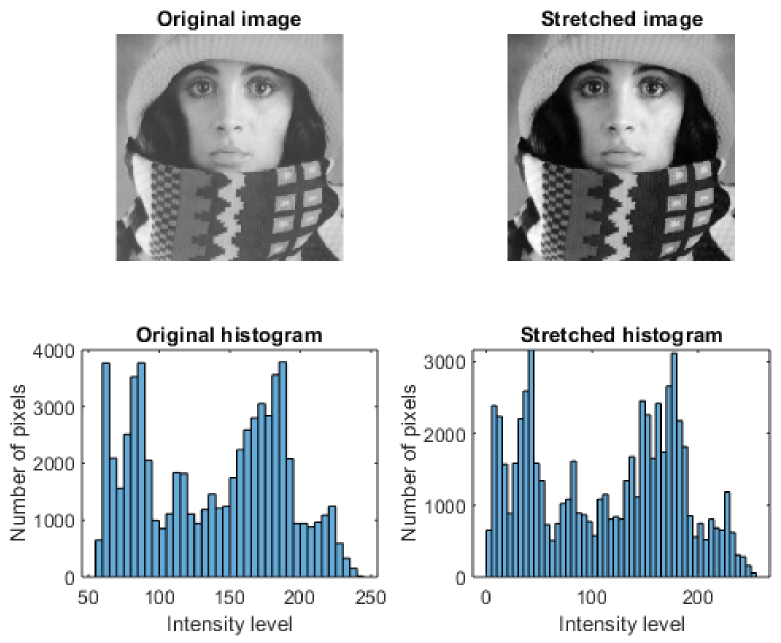


Figure showing original and stretched image with corresponding histograms

c. As a result in this case we get a sharper and brighter image. Observing the histograms there is a clear difference between the original and the contrast stretched image where the contrast stretched image has a larger span of intensity values which includes 0 and 255 whereas this is not the case for the original image.