Q.1 Where in the image is the pixel (1, 1) located and what is the graylevel value? In command window type I(1,1). Did you get the same value?

Right up to the left corner. The Greylevel value is 89.

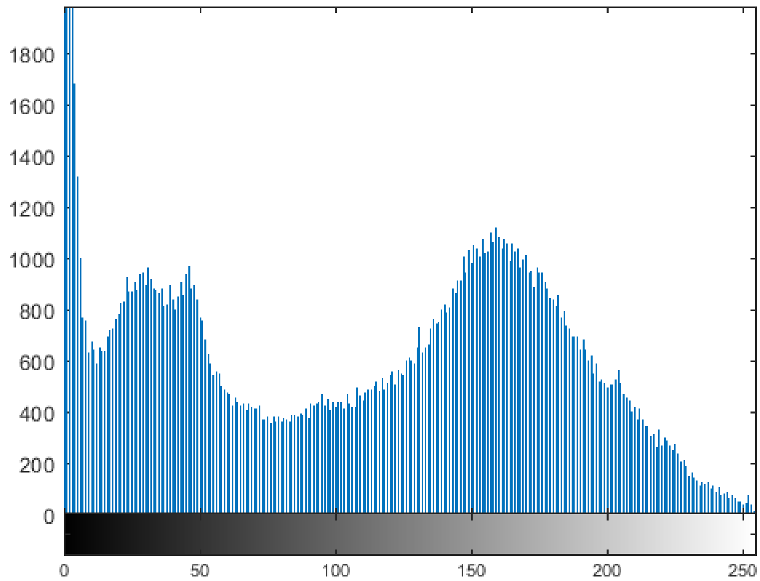
Q.2 Explain what contrast and brightness are and include figures of the histograms of the
images napoleon.png, napoleon light.png, and napoleon dark.png. Can you tell from the histograms which figure has the highest contrast and which figure is the brightest?

Brightness is where on the histogram we put the scale.
- Shifting the histogram to the left will make it darker.
- Shifting the histogram to the right will make it lighter.

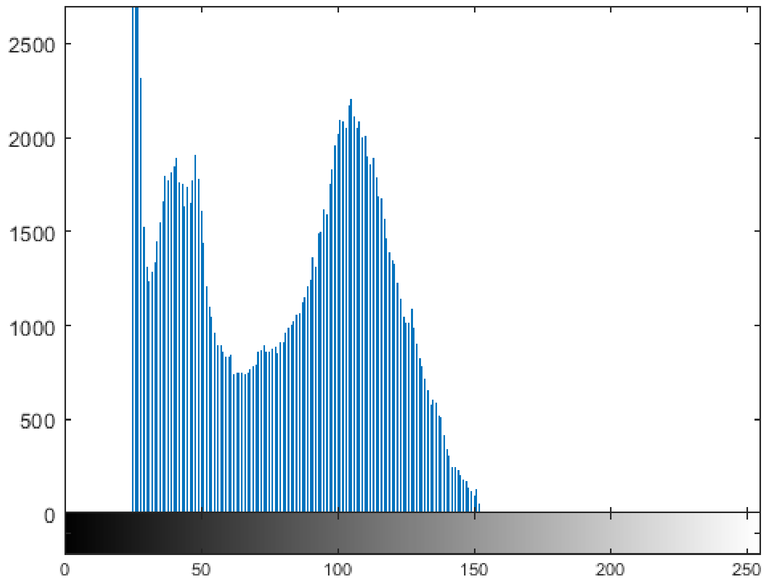
Contrast is how compressed the histograms are. If more compressed → Fewer greylevel values to distinguish the pixels → Lower contrast

The original Napoleon.png photo has the highest contrast.

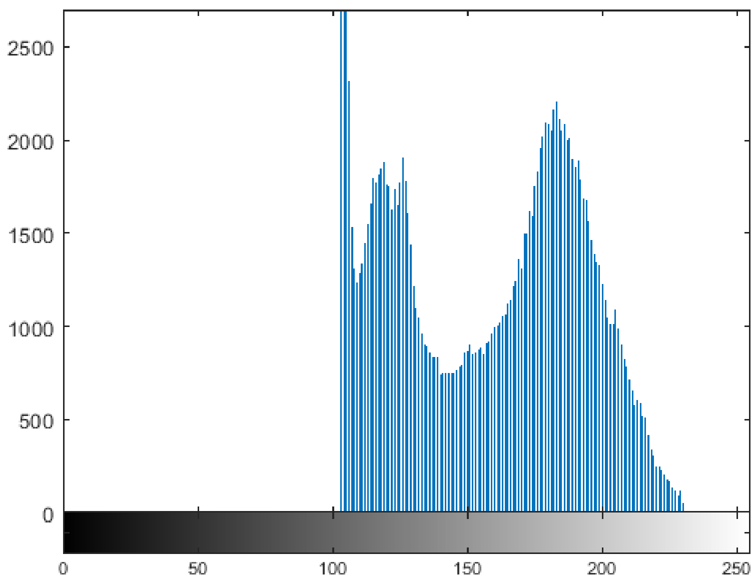
Histogram of Napoleon.png:



Histogram of Napoleon_dark.png:



Histogram of Napoleon_light.png:



Q.3 Explain the difference of imagesc((I/64)*64) and imagesc((Is/64)*64), where I is uint8 and Is is single.

A difference seen in the two figures is that the single produced figure has a higher contrast than the uint8 figure. This is reasonable since the single data type has 32 bits while unit8 only has 8 bits. When decreasing the contrast (by division with 64) information is lost and therefore the original image can't be restored when increasing the contrast (multiplying with 64). Single consists of floats and the uint8 of integers. This affects the values when it get divided by 64, the figure in single data type returns float values and the uint8 rounds the values to an integer. This leads to information loss and difference in the figures' contrast.

Q.4 Demonstrate a mathematical expression involving I to make it brighter.
Adds a constant bigger than 0.

```
I_brighter = I + 30;
```

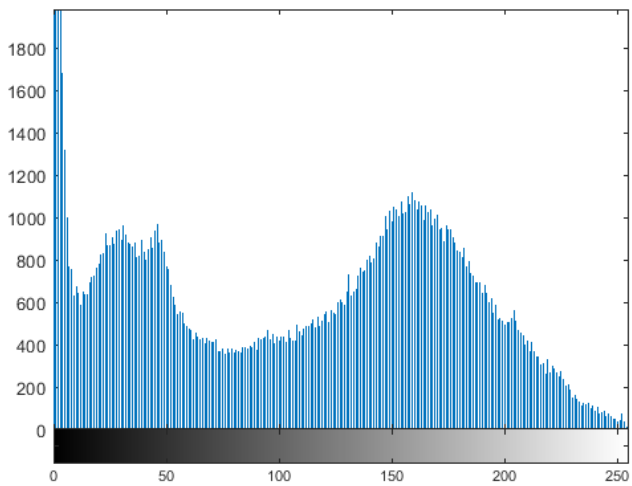
Q.5 Demonstrate a mathematical expression involving I to give it lower contrast.
Multiply with a constant less than 1.

```
I_low_c = I/4;
```

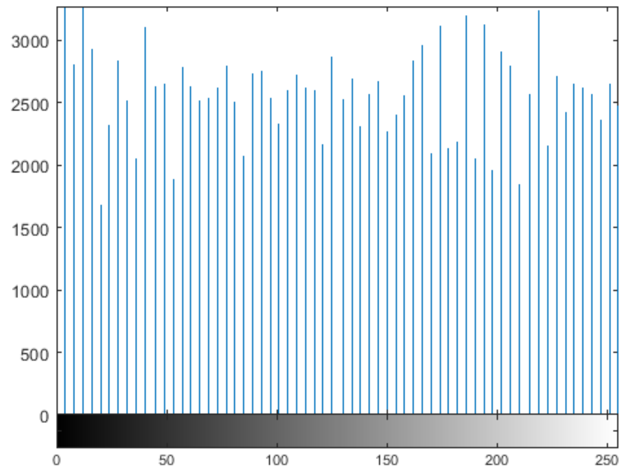
Q.6 Use g = 2 and g = 1/2 and explain the resulting images.
g = 2 → makes the picture darker because it ends up dividing 255 with a bigger value and therefore makes the figure darker (lower values in the greyscale)
g = ½ → makes the picture lighter

Q.7 Explain how histogram equalization works in theory. Include histograms of one of the images before and after applying equalization in your report and explain what you see. Do the changes to the histograms and the images agree with the theory of histogram equalization?

The idea of histogram equalization is to flatten out the histogram. This is done by finding the transformation that produces the most even histogram.
Histogram before equalization:



Histogram after equalization:



The result of a more even histogram can be seen by the figures above where the histogram after the equalization has more similar intensity values of every greyscale value, which agrees with the theory.

Q.8 Explain the role of the interpolation method as well as the role of the lowpass-filter. Which combination of options do you prefer? And why?

Interpolation - Occurs when resizing or distorting image from one pixel grid to another. The lowpass-filter filters out high frequencies and lets the lower ones pass through, in other words the greyscale value with the least intensity passes through.

We prefer the combination of bilinear interpolation and the lowpass-filter. This is because the whole picture is experienced more even and fewer sharp edges, which we think is the purpose of this combination of interpolation and lowpass-filter.

Q.9 Can you give a real-life example of aliasing outside the area of image analysis and signal processing in general? Have you seen this phenomenon before?

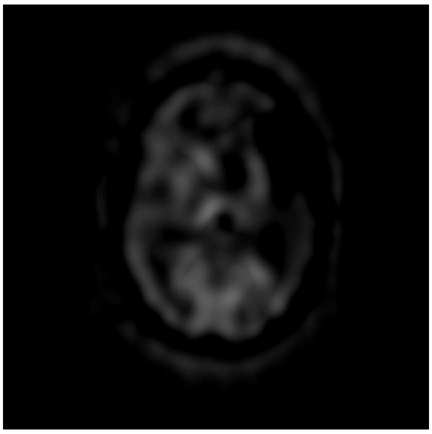
This can be experienced when a car with special types of rims drives by, and the motion for the eye can be seen as going the other way. This is because the experienced frequency is different from the actual frequency, because of the sampling frequency in the brain.

Q.10 How can a "standard" healthy brain, or a mean image, of the two images brain1.png and brain2.png be constructed? In your report include a figure showing the standard brain.

Adding the two images and then dividing, pixelwise, with two (the number of images)

Picture of standard brain:

Q.11 Find the difference between the “standard” brain and the image from the stroke patient (brain3.png). Where in the brain is the change located?
The difference between the standard brain and the stroke patient is found by subtracting the standard brain from the stroke patient’s brain. The major change is located in the bottom of the brain.<
Picture of the stroke patient’s brain:



Q.12 What happens when a pixel gets a value less than 0 or a value greater than 255? Are there other ways this can be handled? Did you think of this when you computed the “standard” brain in the previous exercise?
<0 → Sets to 0
>255 → Sets to 255

Could handle this by shifting the intensities so that the most negative value becomes the zero level. Then we normalize.

Is this a problem when computing the standard brain? Shouldn’t be a problem since the max value should be <= 255

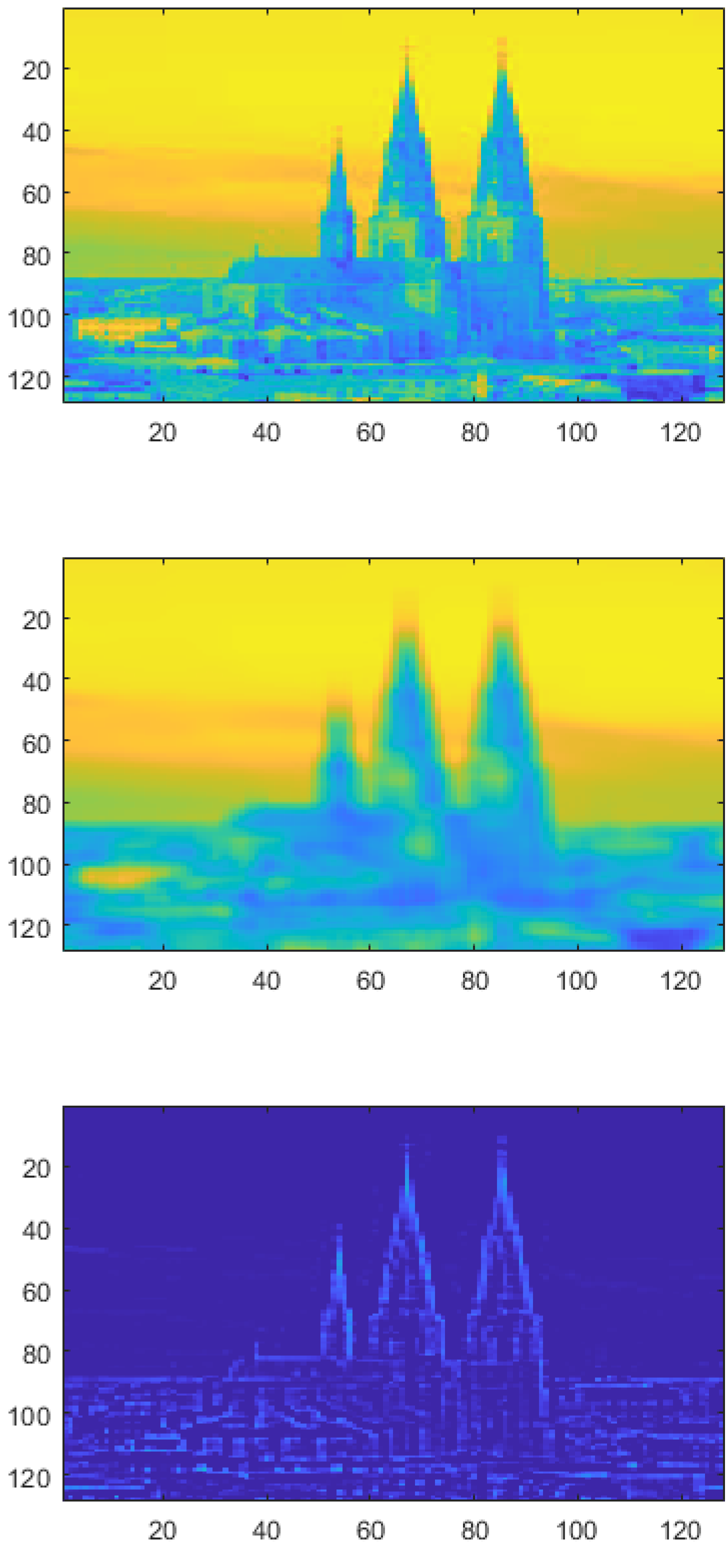
Q.13 Compare rotations performed with and without interpolation. It is easiest to see differences along lines and edges of the images. What does interpolation mean in this case?

The interpolation has a big effect in this case for the edges on the rotated image. When interpolation is used the edges become much more smooth than without, this is because the pixels in the rotated image are of bilinear type which takes a similar to a mean value of the closest pixels. This leads to some gray pixels and looks like a more smooth line for the eye.

Q.14 In general it is faster to rotate the image by a multiple of 90 degrees than by some arbitrary degree. Explain why. For this task use Matlab functions tic and toc.

We think this is faster because the image doesn’t now need to be filled with a black background which affects the interpolarisation. In other words, fewer pixels need to be interpolated which becomes a faster process.

Q.15



20
40
60
80
100
120
20
40
60
80
100
120

Q.16

Our equalized image

Original image

Histeq image

```
function Inew = myhist(I)
   I = imread(I); % Load image

   I = rgb2gray(I);    % Turn into grey-scale

   % Square image
   [x, y] = size(I);

   freq = zeros(1, 256);

   for i = 1:x
       for j = 1:y
           index = I(i, j);
```

```
            freq(1, index) = freq(1, index) + 1;
        end
    end

    nofp = x*y; % number of pixels

    freq_prob = freq./nofp; % Frequency probability

    c = cumsum(freq_prob);  % Cumulative prob distribution

    T = round(c .* 255);   % Mapper

    Inew = T(I);
    Inew = uint8(Inew);
end
```