TNM087 – Image Processing and Analysis
Lab 1 –Intensity Transformations

Preparation tasks

The preparation tasks consist of a few simple problems that should be solved using Matlab. Your answers (in Swedish or English) should be written in the document *Lab_1_Preparation_Answers.doc*, where you also insert the required images. To save the images you can use the Malab functions `imwrite` or `imsave`. Make sure to save the images in an uncompressed format, such as .tif or .png. Before submitting the answer document on Lisam, first save the document as .pdf!

For the preparation tasks you do not need to submit your m-file. However, it is strongly recommended that you save your experiments in an m-file, in case you need to go back and correct anything later. Sometimes, you can also re-use your code in later tasks.

1. Basic image operations and data types

Read the image `'book-cover.tif'` in Matlab, by using the command:
`Image=Imread(‘book-cover.tif’)`
In the workspace you can see some properties for the image, such as the size (rows x columns) and the class (or data type). You can also get these properties by using the command `whos Image`

You see that the Image is of class (or data type) `uint8`, which means that it is stored as 8-bit (1-byte) unsigned integers, in the range 0-255. This will be the case for most images that you will you use. You can display the image using the command `imshow(Image).`

1 A) What is the highest pixel value in the image?

Hint: If you use the command `max(Image)` *for a matrix such as an image, it will return the maximum value for each column. If you want the maximum value for the entire image, you can use* `max(Image(:))`*. The same holds for the function* `min` *(returning the minimum value).*

Now compute a new image with all pixel values in `Image` divided by 16: `Image2=Image/16;`
1 B) What is the maximum value for `Image2`?
1 C) What do you see if you display `Image2`?

Compute a new image, `Image3`, by multiplying `Image2` by a factor 16 and display the new image. What is the result (compared it to the original image)?
1 D) Insert `Image3` in the answer document.
1 E) How many gray levels does `Image3` have?

Hint: You can create a vector `A=uint8(0:255)` *and perform the same operations* `B=(A/16)*16` *to easier see what happen to the pixel values. You can also use the command* `Imhist` *to calculate and display the histogram for the images.*

1 F) Explain what has happened to the image after these operations!

Now convert the original image to the data format `double`, using `Image=double(Image)/255;` or `Image=im2double(Image);`.

Images in `double` class are stored with 64 bits precision and should be in the range [0,1] This is the reason for dividing the uint8 image by 255 when using `double`. This is automatically handled by using `im2double`.

Perform the same operations for the image in `double` format and observe the result.
1 G) Explain the difference between using `uint8` images and `double` images in this task.

In the following tasks (as well as in most laboratory tasks in the course) you will often use arithmetic operations (such as division, multiplication, subtraction and addition) on images.
1 H) Which class (data type) should you make sure to use when applying such operations to images?

2. Contrast stretching and image histogram

Read, convert to `double`, and display the image ‘`einstein-low-contrast.tif`’
What can you say about the image?
2 A) What is the max- and min- values for the image?

Try to predict how the histogram for the image will look, then calculate and view the histogram using the command `imhist`.
2 B) Insert the histogram in the answer document

A common way to expand the range of intensity levels in images with low contrast is to use contrast stretching (sometimes called autolevel), as:

$$g_s(x,y) = K\frac{g(x,y)-g_{min}}{g_{max}-g_{min}}$$

where g_{min} and g_{max} are the min- and max-values for the image *g(x,y)* and *K* is a constant for scaling the stretched image to the range [0,*K*] (i.e. *K*=1 or *K*=255 for images in `double` or `uint8` format, respectively)

Perform contrast stretching on your low contrast image and view the result.
2 C) Insert the resulting image in the answer document.

2 D) What will the max- and min- values be for the stretched image? (You can predict the result by just looking at the equation!)

2 E) View the histogram for the stretched image and insert it in the answer document.

3. Image subtraction

It is sometimes useful to use image subtraction to display the difference to a reference image, often called a *mask*. The image `’angiography-mask-image.tif'` is an X-ray image of a patient’s head that serves as a mask. The image `’angiography-live-image.tif'`is another X-ray image, after injecting a contrast medium (see sec. 2.6 in the textbook for details about the images). By just viewing the two images side by side it is not trivial to identify how the contrast medium propagates through the blood stream. Read and view the two images in Matlab!

To easier follow the contrast medium, use `’angiography-mask-image.tif'`as a mask image *h*(*x*,*y*) and compute the difference image: *g*(*x*,*y*) = *f*(*x*,*y*)- *h*(*x*,*y*).

Hint: If your images are in uint8 format, what will happen to any negative values after the subtraction?

A difference image such as $g(x,y)$ will generally contain negative values and often be of low contrast. Use contrast stretching to map $g(x,y)$ to the optimal range and display the result!

3 A) Insert the enhanced difference image in the answer document

Hint: If you use the command `imshow(Image,[])` *in Matlab, the lowest value (which can be negative) will automatically be displayed as black and the highest value in the image will be displayed as white. The effect is the same as displaying an image after contrast stretching but without changing the image data, which can sometimes be useful.*

4. Histogram equalization

Read and display the image `'pollen-lowcontrast.tif'`
What can you say about the image?
Try to predict the histogram for the image, then calculate and view the histogram.

Now perform histogram equalization on the image, using the command `histeq`.
Display the resulting image and its histogram. What can you say about the resulting image? Does the histogram for the equalized image look as you expected?
4 A) Insert the equalized image in the answer document
4 B) Insert the histogram for the equalized image in the answer document

5. Image division and shading correction

The image `'Shade_pattern.tif'` contains a shaded test pattern (the shade could for example be the result of a non-uniform illumination). Display the image, and its histogram. Your goal is to separate the white and black squares in the test pattern, by finding a threshold.
5 A) Insert the histogram in the answer document
5 B) Is it possible to find a global threshold to segment this image (look at the histogram)?

A simple model for image shading is: $g(x,y) = f(x,y)\, h(x,y)$,
where $g(x,y)$ is the resulting shaded image (e.g. `'Shade_pattern.tif'`),
$f(x,y)$ is the “perfect image” without shading, and $h(x,y)$ is the shading function.

If the shading function $h(x,y)$ can be approximated, the undistorted image $f(x,y)$ can be estimated by dividing $g(x,y)$ by $h(x,y)$, using elementwise division. Luckily, you have an approximation of the shading function $h(x,y)$ in the image `'Shade_estimate.tif'`. Use it to remove the shading from `'Shade_pattern.tif'`!

Hint: using image division (and multiplication) is elementwise operations, i.e. it is carried out on a pixel-by-pixel basis. To perform elementwise operations in Matlab you must use `./` *or* `.*` *(if you use only* `/` *or* `*` *you will instead perform matrix operations). Never use for-loops for elementwise operations!*

5 C) Insert your recovered image in the answer document
5 D) Insert the histogram of recovered image in the answer document

Look at the histogram to find a threshold to segment the black squares from the white.
5 E) Segment the image (using the threshold) and insert the result in the answer document
5 F) What is the data type (class) for the segmented image?
5 G) How many bits (per pixel) is required to store this type of image?

Hint: the Matlab command `BW =imbinarize(Image,T)` *creates a binary image using the specified threshold value, T. But for simple global thresholding, you can simply use:* `BW=Image>T`*.*

6. RGB-images and indexing

Create an empty matrix: `I = zeros(400,600,3);`
The matrix, `I`, has the dimensions 400 rows x 600 columns x 3 channels, and can thus be considered an RGB-image (with the 3 channels corresponding to red, green and blue). The data type is double and since it consists of zeros only, it will show as black. If you use `I(100:200,:,1)=1;` to modify the matrix, you will create a vertical band of red pixels (were the 1st channel set to 1), covering the rows 100-200 (counting from the top of the image), and all the columns. In the same way, `I(:,200:400,2)=1;` will create a green vertical band, covering all rows and the columns 200:400. `I(150:350,300:500,3)=1;` will create a square in the blue channel, between the rows 150:350 and the columns 300:500. The parts of the image were 2 channels overlaps, you will see the secondary colors (cyan, magenta and yellow), and the parts were all 3 channels are set to 1 you will have a white area. Try out these operations, display the resulting color images, and make sure that you understand how the indexing works.

In the same way (i.e. creating a 400 x 600 x 3 matrix and modifying the 3 color channels using matrix indexing only), now create a Swedish flag!

6 A) Insert the flag in the answer document!