Image Analysis, Assignment 1

1 Image sampling

Consider a continuous monochrome image. The intensity (brightness) in a point (x, y) in the image is given by the function

$$f(x, y) = x(1-y) + y(1-x), \quad 0 \leq x \leq 1, 0 \leq y \leq 1.$$

Think about what the image looks like.

Sample the image evenly to a discrete image with 5×5 pixels. Let the lower left pixel be a sample from the point $(0, 0)$ in the continuous image and the upper right pixel a sample from $(1, 1)$. Quantify the discrete image with 16 different gray levels from 0 to 15. What is the result?

For the report: Write out the resulting 5×5 image matrix. Explain how you did your calculations.

2 Histogram equalization

An image (in a continuous representation) has gray level histogram

$$p_r = \frac{3}{2}\sqrt{r}, \quad r \in [0, 1] \ .$$

What gray level transform $s = T(r)$ should be used so that the resulting histogram p_s is uniform, i.e.

$$p_s = 1, \quad s \in [0, 1] \ ?$$

For the report: Specify the transformation $s = T(r)$ and show how you computed it.

3 Neighbourhood of pixels

Consider the following image

$$\begin{pmatrix} 3 & 3 & 2 & 2 & 1 & 3 & 3 & 3 & 0 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 3 & 2 & 3 \\ 1 & 2 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 0 & 2 & 3 \\ 3 & 0 & 1 & 1 & 0 & 3 & 3 & 1 & 2 & 3 & 3 & 2 \\ 3 & 2 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 0 & 3 & 2 \\ 0 & 3 & 3 & 1 & 1 & 1 & 0 & 0 & 1 & 3 & 3 & 3 \\ 2 & 0 & 2 & 1 & 1 & 0 & 1 & 0 & 3 & 2 & 0 & 2 \\ 2 & 1 & 3 & 0 & 2 & 3 & 2 & 1 & 2 & 0 & 3 & 2 \\ 3 & 0 & 0 & 1 & 3 & 3 & 2 & 0 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & 2 & 1 & 1 & 2 & 0 & 1 & 0 & 2 & 1 \\ 0 & 2 & 3 & 0 & 0 & 1 & 0 & 0 & 1 & 3 & 3 & 2 \\ 0 & 2 & 3 & 0 & 3 & 3 & 3 & 0 & 2 & 1 & 2 & 2 \end{pmatrix}.$$

a) Mark each element that has intensity 0 or 1 with a circle.

For a pixel with coordinates (m, n) the 4-neighbours (as defined in lecture 1) are the four pixels with coordinates $(m \pm 1, n)$ and $(m, n \pm 1)$.

b) Fill the circles of those pixels that have intensity 0 or 1 and have **at least two** 4-neighbours that also have intensities 0 or 1.

For the report: Perhaps the easiest way to do this is by hand, e.g. by taking a photo of your solution and putting it in the report.

4 Segmentation part of OCR

During all of the four assignments you are gradually going to build and test a small system for ocr (optical character recognition). During this first assignment your task is to write one function S = im2segment(image) in matlab. And to test this function on a few images using a benchmark script `inl1_test_and_benchmark`.

On the web page for the course in image analysis there is a zip-file,
`inl1_to_students.zip`
By downloading and unpacking the file you will obtain a folder `ocr_project` and in this folder there are two folders
`datasets`
and
`matlab`
In the datasets folder there is for now only one folder 'short1', which contains a few test examples and ground-truth both for segmentation and for recognition. Later on we will add more folders with additional (and more challenging images).

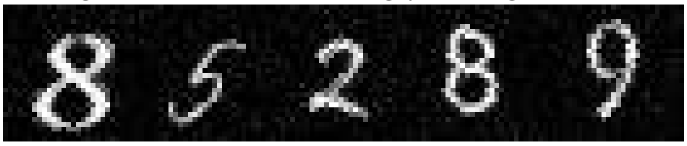
Study the script `inl1_stub.m`, that reads one of the images in the folder short1. Each image in the folder contains text (light against dark background) Write a matlab program `im2segment` that takes such an image matrix I as input and returns a segmentation, i.e.

a set of images $S = (S_1, \ldots, S_n)$ one for each digit in the image. Each such image matrix S_i should be a matrix with ones at the pixels for that digit and zeros for all other pixels. In order to make things easy for all of us, you should all use the same convention to let the output be a so called cell array in matlab.

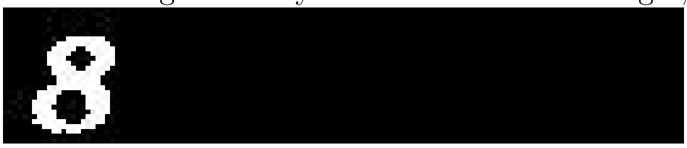
```
S = cell(1,n);
S{1} = bild1;
...
S{n} = bildn;
```

Also make sure that each image (e.g. `bild1`) has the same size as the original image.

If all goes well when running your segmentation on the follwing image (im)



the resulting cell array will then contain 5 images, i.e. `S{1}` is



the second segment `S{2}` is

and so on. Again notice that each segmented image is of the same size as the original image.

Test your function `im2segment` using the benchmark script `inl1_test_and_benchmark.m` that loads each image in a folder and each ground truth segmentation and measures how well it works.

For the report:

- Print your code im2segment.m and the text results of running the benchmark script.
- Include a figure of one example of an input image and the resulting segmentation images, similar to the ones above.
- Make sure that the code is commented so that it is easy to follow, or write a description of what it does.

5 Dimensionality

A **vector space** is a collection of objects and two operations **addition** and **multiplication by scalar**. For finite dimensional vector spaces one may choose a basis of elements $e_1, \ldots, e_k$ so that every example u can be written as

$$u = x_1 e_1 + \ldots, +x_k e_k$$

for some set of scalars $(x_1, \ldots, x_k)$. Here k is the dimension of the vector space.

A: The set of gray-scale (monochrome) images with 2×2 pixels is a vector space. It has a finite dimension k, and we can define a basis that consists of k 2×2 images.

B: The set of gray-scale (monochrome) images with 2000×3000 pixels is a vector space.

For the report:

- In A what is the dimension k?
- Give an example basis for A (by explicitly writing down the basis elements $e_1, \ldots, e_k$).
- In B what is the dimension k?
- In B describe how the basis elements can be chosen.

6 Scalar products and norm on images

Given three images

$$u = \begin{bmatrix} 6 & -2 \\ 1 & -3 \end{bmatrix},$$

$$v = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix},$$

$$w = \frac{1}{2}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad .$$

For the report:

- How is the scalar product defined for images?
- How is the norm of an image defined?
- Calculate $||u||$, $||v||$, $||w||$, $u \cdot v$, $u \cdot w$, $v \cdot w$.
- Are the matrices $\{v, w\}$ orthonormal?
- What is the orthogonal projection of u on the subspace spanned by $\{v, w\}$? Is the resulting projection a good approximation?

(Notice that there are many possible matrix norms! There is a potential pitfall here if you use a computer program, e.g. matlab. You might accidentally be using the wrong norm!)

7 Image compression

A small camera delivers low resolution images with 3×4 pixels. Before transmitting the image to a computer, one would like to compress the images consisting of 12 intensities to 4 numbers. After studying numerous images and using principal component analysis one has determined that the following four images represent typical images well,

$$\phi_1 = \frac{1}{3}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \phi_2 = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \phi_3 = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \phi_4 = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}.$$

Assume that we have an original image

$$f = \begin{pmatrix} -2 & 6 & 3 \\ 13 & 7 & 5 \\ 7 & 1 & 8 \\ -3 & 3 & 4 \end{pmatrix} .$$

We would like to find an approximation f_a of f using the four basis images.

For the report:

- Show that the four images $\phi_1, \phi_2, \phi_3, \phi_4$ are orthonormal (write out all combinations you test, and their result).
- How do you determine the four numbers (coordinates) x_1, x_2, x_3, x_4 such that the approximate image
$$f_a = x_1\phi_1 + x_2\phi_2 + x_3\phi_3 + x_4\phi_4$$
is as close to f as possible, i.e. such that $|f - f_a|^2 = (f - f_a) \cdot (f - f_a)$ is as small as possible?
- Calculate x_1, x_2, x_3, x_4 for the image f above.
- Determine the approximate f_a.
- Is f_a close to f?

8 Image bases

On the web page for the course there is a file, zip-file, `assignment1bases.mat` with two variables `bases` and `stacks`.

The variable `stacks` is a cell array. It contains two stacks of images

- 400 test images of faces of size 19×19. These are stored in a variable `stacks{1}`, which is a three-dimensional data structure of size $19 \times 19 \times 400$.
- 400 other test images of size 19×19. These are stored in a variable `stacks{2}`, which is a three-dimensional data structure of size $19 \times 19 \times 400$.

In the previous exercise we saw how to project an image onto a low-dimensional subspace defined by a set of basis images. Some interesting questions for discussion are:

- Is there a difference between different bases?
- What is the best basis?
- How can one calculate a good basis?

We do not expect a full answer to these three questions, but we encourage you to think about them and to discuss them with your fellow students.

The variable `bases` is also a cell array. It contains three sets of bases for three different subspaces of dimension four. The first basis is stored in a variable `bases{1}`, which is a tensor of size $19 \times 19 \times 4$. Thus the four basis images are `bases{1}(:,:,1)`, `bases{1}(:,:,2)`, `bases{1}(:,:,3)`, `bases{1}(:,:,4)`.

Write a matlab function that projects an image u onto a basis (e_1, e_2, e_3, e_4) and returns the projection u_p and error norm r, i.e. the norm of the difference $r = |u - u_p|$.

Then write a script that tests all of the 400 test images in a test set and returns the mean of the error norms. Calculate this mean for each of the two test sets on each of the three bases.

Discussion? How should one calculate the best basis for a stack of images?

For the report:

- Include the code of your matlab function for projection, with explanation of how it works.
- plot a few images in each of the two test sets.
- describe in your own words what the images look like in the two test sets.
- Include plots of the four basis elements of each of the three bases and describe shortly the visual differences between the three bases.
- Print the mean of the error norms for the six combinations (two test sets against the three bases).
- Which basis works best for test set 1? Why?
- Which basis works best for test set 2? Why?