Image Analysis
Exercise 06 - Blob Analysis and cell counting

2019

Introduction

The purpose of this exercise is to experiment with connected component analysis, also called blob analysis.

As usual you start by creating an empty M-file:

```
edit BlobAnalysis.m
```

Data

The data and material needed for this exercise can be downloaded from

`http://courses.compute.dtu.dk/02502/`.

Labelling

In Matlab, the function `bwlabel` is used for labelling a binary image. Find out how this function works using the help command.

```
doc bwlabel
```

We start by working with a small synthetic image. Load the image and display it (make sure you have `imagegrid` in the Matlab directory).

```
load Image1.mat
imagesc(Image1);
imagegrid(gca,size(Image1));
colormap(gca, hot);
```

Exercise 1 *Apply* `bwlabel` *on Image1 using 4 neighbours:*

```
L4 = bwlabel(Image1, 4);
```

`L4` *is a new image. Show it together with the original using the* `subplot` *command.*

Exercise 2 *Apply* `bwlabel` *on Image1 using 8 neighbours. Show it together with the original and the result of the 4-neighbour operation using the* `subplot` *command.*

Explain the difference in the output.

It can be difficult to see the different label colours. This can be fixed using `label2rgb`. A colour-label image is created for example by:

```
RGB4 = label2rgb(L4);
```

or

```
RGB4 = label2rgb(L4, 'spring', 'c', 'shuffle');
```

Exercise 3 *Use* `label2rgb` *to make the figures from the previous exercise more clear.*

Object properties

The Matlab function `regionprops` allows you to get different summary values of the labelled components. It returns an array of structs.

Exercise 4 *Start by calculating the area properties of the 8-connected components*

```
stats8 = regionprops(L8, 'Area');
```

To inspect the value of an element, for example the area of the first blob, the following can be used:

```
val = stats8(1).Area;
```

Exercise 5 *What are the areas of the three first blobs?*

It is more efficient to get all the measured areas out in one vector. This can be done like:

```
allArea = [stats8.Area];
```

Exercise 6 *Make a list of all the measured areas in the image.*

The function `ismember` is useful in conjunction with `regionprops` for selecting regions based on certain criteria. For example, these commands create a binary image containing only the regions whose area is greater than 16.

```
 idx = find([stats8.Area] > 16);
 BW2 = ismember(L8,idx);
```

Exercise 7 *Show an image where only the blobs with area larger than 16 is shown.*

As seen in the documentation for `regionprops` many different properties can be extracted.

Exercise 8 *Make a list of all the perimeters of the objects.*

Note that there are several ways of computing the perimeter of an object. The perimeter computed by `regionprops` is different from described in the textbook. There will not be exam questions directly asking you to compute the perimeter of an object.

Exercise 9 *How many objects have a perimeter larger than 20? (use for example* `numel` *to count)*

Exercise 10 *Plot the object perimeters as function of object areas. Use for example:*

```
  plot(allArea, allPerimeter, '*');
```

Play around with the plot so it become as you want. Is there a high correlation between the two measurements?

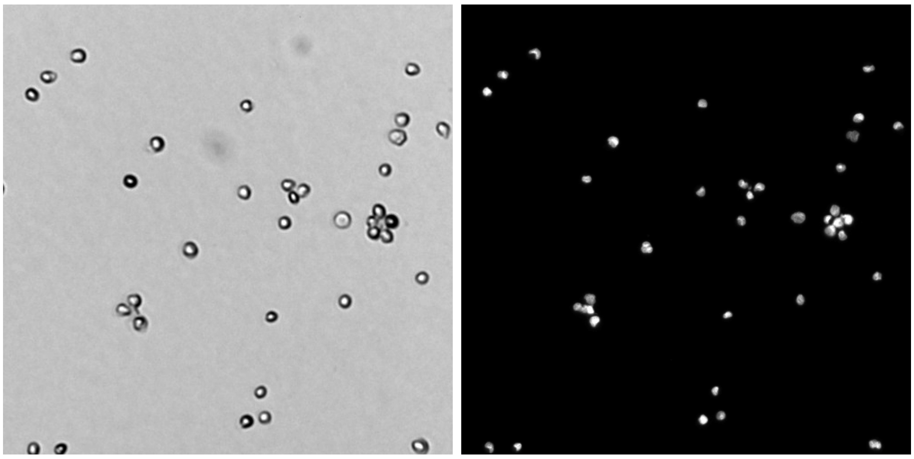
Cell counting

In this exercise the goal is to create a Matlab program that can count cell nuclei in microscopy images.

The images used for the exercise is acquired by the Danish company Chemometec using their image-based cytometers. A cytometer is a machine used in many laboratories to do automated cell counting and analysis. An example image can be seen in Figure 1 where U2OS cells (human bone cells) have been imaged using ultraviolet (UV) microscopy and a fluorescent staining method named DAPI. Using DAPI staining only the cell nuclei are visible which makes the method very suitable for cell counting.

The raw images from the Cytometer are 1920x1440 pixels and each pixel is 16 bit (values from 0 to 65535). The resolution is $1.11\mu m$ / pixel.

To make it easier to develop the cell counting program we start by working with smaller areas of the raw images. The images are also converted to 8 bit grayscale images:



Figur 1: U2OS cells. To the left image acquired using UV microscopy and to the right the corresponding DAPI image.

```
clear all,close all,clc; % Clean the workspace
I16 = imread('CellData\Sample E2 - U2OS DAPI channel.tiff');
I16c = imcrop(I16, [700 900 500 500]); % Crop region from raw image
Im = im2uint8(I16c); % Convert region into 8-bit grayscale
imshow(Im, [0 150]); title('DAPI Stained U2OS cell nuclei');
```

Exercise 11 *Read the documentation for* `imcrop` *so you can use it later for extracting relevant regions from the raw image.*

Exercise 12 *As can be seen* `imshow` *can be used to visualise specific gray scale ranges (0 to 150 above). Adjust these limits to find out where the cell nuclei are most visible.*

We start the analysis by creating a binary version of `Im` where cell nuclei become foreground.

Exercise 13 *Examine the histogram of the image (*`imhist`*) and select an appropriate threshold. Apply the threshold and create a binary image called BW. Show it.*

It can be seen that there is some noise (non-nuclei) present and that some nuclei are connected. Nuclei that are overlapping very much should be discarded in the analysis. However, if they are only touching each other a little we can try to separate them. More on this later.

To make the following analysis easier the objects that touches the border should be removed.

Exercise 14 *Read the documentation for* `imclearborder` *and use it to create a new binary image where border objects are removed.*

To be able to analyse the individual objects, the objects should be labelled.

Exercise 15 *Use the methods from the labelling section to create a label image using 8-neighbourhood.*

The task is now to find some *object properties* that identifies the cell nuclei and let us remove noise and connected nuclei.

First we can see if the area is enough

Exercise 16 *Use* `regionprops` *to compute the area of all the objects in the binary image where the border objects are removed.*

Plot a histogram of all the areas and see if it can be used to identify well separated nuclei from overlapping nuclei and noise. Look at the documentation of `hist` *to see how you can change the number of bins.*

Exercise 17 *Try to select a minimum and maximum allowed area and use* `ismember` *to extract the cells and show them in a new image. Also count the number of accepted objects. Did you succeed in finding the good cell and removing non-cell objects?*

We should also examine if the shape of the cells can identify them. A good measure of how circular an object is can be computed as:

$$f_{\text{circ}} = \frac{2\sqrt{\pi A}}{P}, \tag{1}$$

where A is the object area and P is the perimeter.

Exercise 18 *Use the area and perimeter from* `regionprops` *to compute* f_{circ} *for all objects.*

Show a histogram of f_{circ} *and see if there are some obvious minimum and maximum values that can be used to identify good cells.*

Remember that you can combine the measured values when extracting the objects.

Exercise 19 *Use a combination of area and* f_{circ} *to select good cells.*

Exercise 20 *Create and implement a new function*

```
function [I,N] = CountCellNuclei(Im)
% CountCellNuclei Count the number of cell nuclei in an image
% specially designed for DAPI stained images from Chemometec
% Return an image (I) with cells nuclei and the number of nuclei N
```

The function should accept an 8-bit grayscale image like in the DAPI stained image in Figure 1. It should return a binary image with the cell nuclei and the number of found nuclei.

Exercise 21 *Test your CountCellNuclei function on a larger set of training images. In the table below the suggested training images and regions are seen. Use* `imcrop` *to select the correct regions from the raw image. You can also try it on regions that you select yourself.*

File	Cell Type	Selection (x,y,width,heigh)
Sample E2 - U2OS DAPI channel	U2OS	700, 900, 500, 500
Sample E2 - U2OS DAPI channel	U2OS	0, 0, 500, 500
Sample E2 - U2OS DAPI channel	U2OS	600, 200, 500, 500
Sample E2 - U2OS DAPI channel	U2OS	1300, 0, 500, 500
Sample E2 - U2OS DAPI channel	U2OS	900, 500, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	0, 0, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	500, 0, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	0, 500, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	500, 500, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	1000, 0, 500, 500
Sample G1 - COS7 cells DAPI channel	COS7	1000, 500, 500, 500

COS7 cells are *African Green Monkey Fibroblast-like Kidney Cells* (www.cos-7.com) used for a variety of research purposes.

Exercise 22 *Evaluate the results of you method based on the examples. Did you manage to successfully separate cell nuclei from noise?*

Exercise 23 *Try to adjust the minimum and maximum values that was used in the selection. Also try other measures from* `regionprobs` *to see how good your algorithm can become.*

Handling overlap

In certain cases cell nuclei are touching and are therefore being treated as one object. It can sometimes be solved using for example the morphological operation opening before the object labelling. The operation erosion can also be used but it changes the object area.

Exercise 24 *Experiment with morphological operations to see if you can separate touching cells before counting them.*