Application Note

Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

Counting of yeast colonies in Petri dish images

Jan Schier
schier@utia.cas.cz, +420-2-6605 2470

Centre for Applied Cybernetics
1M0567

Department of Signal Processing
http://zs.utia.cz

Contents

Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	18.08.2009	J.S.	Vytvoření dokumentu

1 Introduction

This document describes processing flow used for counting of yeast colonies, performed as a part of quantitative analysis of images of the colonies growing on a Petri dish. It is accompanied by Matlab demo, showing the results of counting on several sample images of Petri dishes.

Quantitative analysis of yeast growth is used in some experiments to determine the influence of a substance, contained in the growth medium, on the growth of the colonies. The colonies grow on a solid medium contained in a Petri dish, which is stored in a cultivation box.

The demo is based on two functions: `Counting()`, which computes the estimate of the number of colonies contained in the dish, and `DishPreprocessor()`, which performs the necessary image preprocessing (adaptive image thresholding to eliminate the background, locating the dish and error checking). The functionality of `DishPreprocessor()` has been described in detail in [1]. This report covers in detail the internals of the `Counting()` function.

Let us note that the demo is extracted from a software tool that integrates both the preprocessing and area/colony counting and an image editor used to adjust the inaccuracies in the results of colony counting (Figure 1). For more information on this tool, do not hesitate to contact our group at `schier@utia.cas.cz`. We are interested in cooperation with possible users for further development of the tool, and, especially, for preparation of joint research projects. A version of the tool based on non-commercial implementation tools is planned, we are, however, not able to provide any release date.

The images included in the demo have been prepared in the Department of Genetics and Microbiology, Faculty of Natural Sciences, Charles University. The software has been developed in cooperation with this laboratory.

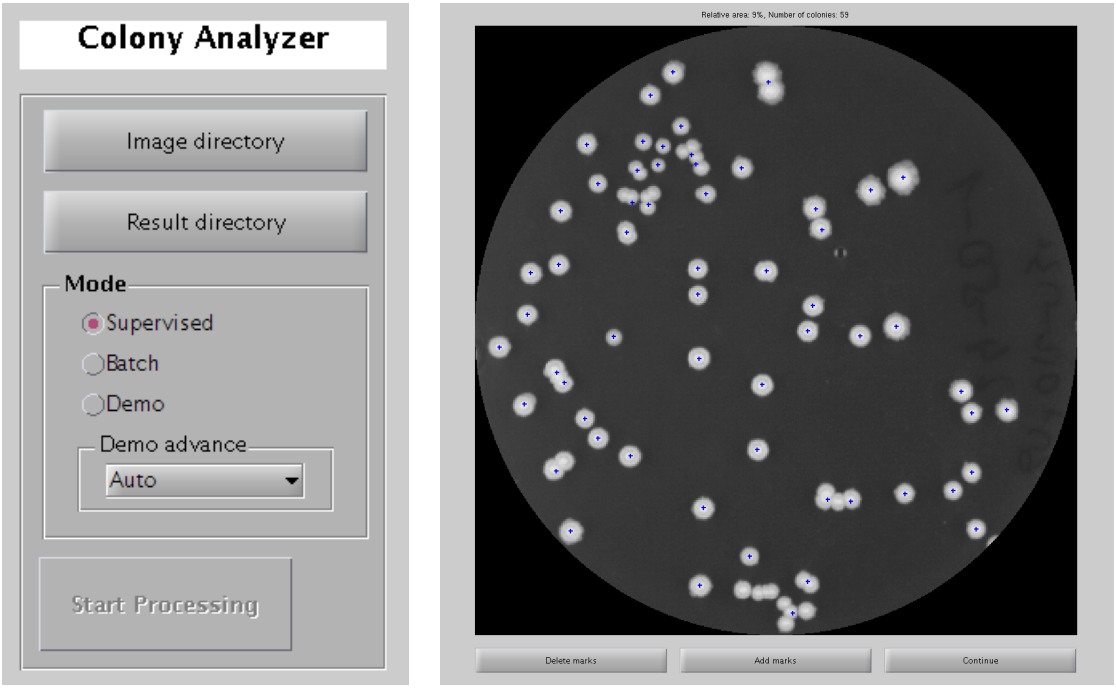


Figure 1: Colony analyzer tool

1.1 Structure of the report

The report is structured in the following way:

- first, the usage of the demo and of its main function `Counting()` are explained,

- then, the morphological features of the yeast colonies images are outlined,
- the processing flow implemented in the `Counting()` function is described,
- finally, the results of processing several test images are given.

2 Demo description

2.1 Software requirements

The demo is tested with Matlab (R) 7.7.0 (R2008b) and is expected to run also with any newer version of Matlab. It requires the Image processing toolbox (TM).

2.2 Using the demo

The demo is located in the directory `Demo`. To run it, start Matlab and change the working directory to `Demo`. Then, type `demo` with the name of the image file to be processed as parameter:

```
>> demo('SampleImg/sample1.jpg')
```

in Linux or

```
>> demo('SampleImg\sample1.jpg')
```

in MS Windows.

Demo reads the file sample1.jpg, and displays the output: the original gray-scale image of the dish with a circle marking the region of interest (ROI) and a mark showing the center of the dish, and a second image showing only the colonies in gray-scale, the circle marking the region where the area of the dish is considered (it is the same circle as in the first image) and cross-signs marking the colonies, as estimated by the function. See Figure 2.

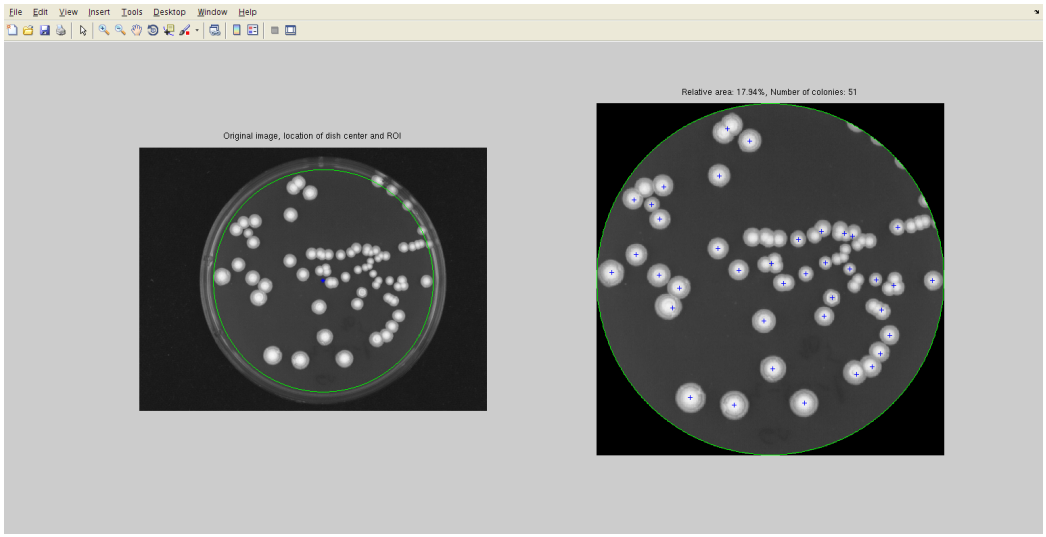


Figure 2: Demo output

Internally, the demo calls the function `DishPreprocessor()`, with the matrix `img` containing the gray-scale image as parameter:

```
[Dish,DishBW,PetriX,PetriY,R_ROI,TotArea] = DishPreprocessor(img);
```

The call of `DishPreprocessor()` is immediately followed by the call of the `Counting()` function.

```
[area, NBlobs, BlobCenter] = Counting(Dish, DishBW, TotArea);
```

The parameters Dish, DishBW and TotArea, which are passed between the two functions, have the following meaning:

Dish, DishBW	gray-scale and binary ROI image (the `Dish` matrix contains also the Petri dish background)
TotArea	total ROI area (used to compute the relative area of the colonies)

The output parameters `area`, `NBlobs`, `BlobCenter` of the `Counting()` function have this meaning:

area	Relative area of the colonies
NBlobs	Estimated number of blobs (colonies)
BlobCenter	Coordinates of the blob centers

The internals of the `Counting()` function will be described in Section 4. The functionality of `DishPreprocessor()` has been described in [1].

3 Image characteristics

For each dish, images are taken several several times during the growth of the colonies. Images are taken in a dark room, using digital camera with two light sources mounted on a general-purpose imaging mount by Kaiser Fototechnik (see Figure 3). The colonies are roughly round-shaped, there are, however,

Figure 3: Imaging configuration

large variations in size, morphology (the surface can be either smooth or fluffy), and the exact shape, which can be nearly circular or rather fuzzy, dependent on the age of colonies and the tested substance in the medium. Furthermore, several colonies may be touching each other.

Several typical examples are given in Figure 4.

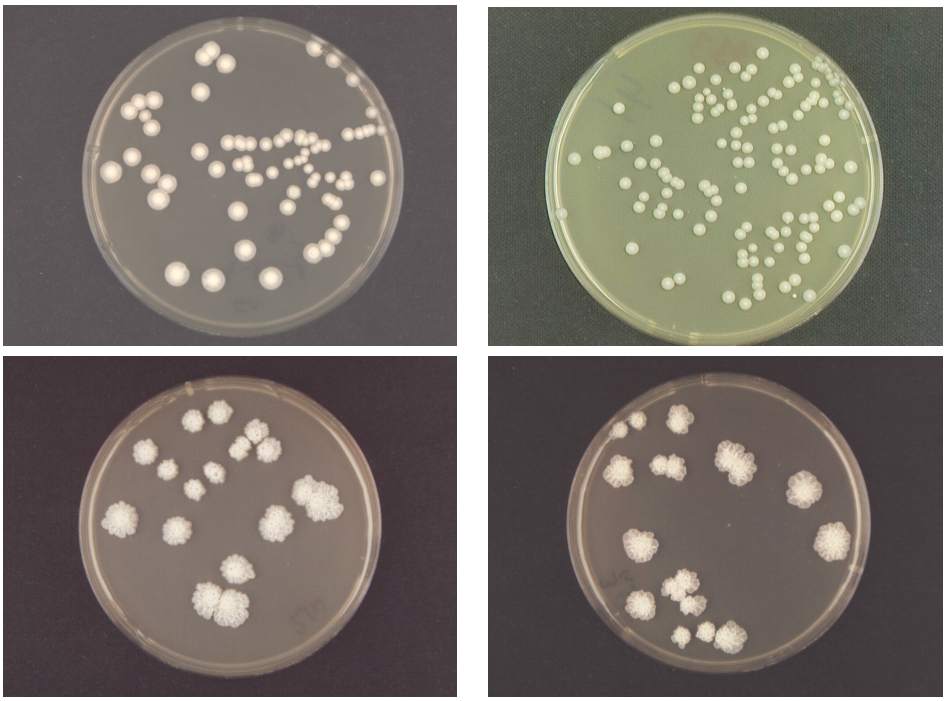


Figure 4: Examples of yeast colonies

4 Processing flow of counting of the colonies

In this section, the processing flow used in the `Counting()` function will be described in detail. The flow is summarized in the chart presented in Figure 5.

To demonstrate the function of each block in the chart, an example image as shown in Figure 6 is used (note: it's the same image as shown in Figure 1 and 2.

In the following text, the function of the blocks used in the `Counting()` function will be discussed.

4.1 Noise speckle filtering

This operation is used to eliminate the noise speckles in the image. Two iterations of image erosion on binary image of the colonies are used:

```
DishEroded = imerode(DishBW,Mask);
DishEroded = imerode(DishEroded,Mask);
```

4.2 Rim-based blob separation

This operation is used as the first step to separate touching colonies based on the grey level of their rim, which is often lower then the grey level of the center (as shown in Figure 7). The aim is to set the image threshold between the grey level of the colony rim and the level of the colony center.

The principle of the method is outlined in Figure 8: an eroded B/W image of the dish is used to mask the centers of the colonies in the grayscale image. A histogram is computed for the masked image, containing only the rim of the colonies. The position of the maximum of this histogram is used to set the threshold for conversion of the image to binary form. Finally, image filling and a 2-step erosion are used.

For the image used as the example, there were 66 distinct blobs (both single and touching colonies) before using the rim-based separation. After the separation, 73 blobs were recognized. Using only image erosion, without the histogram-based threshold, the results were: after 2 repetitions, 67 blobs were

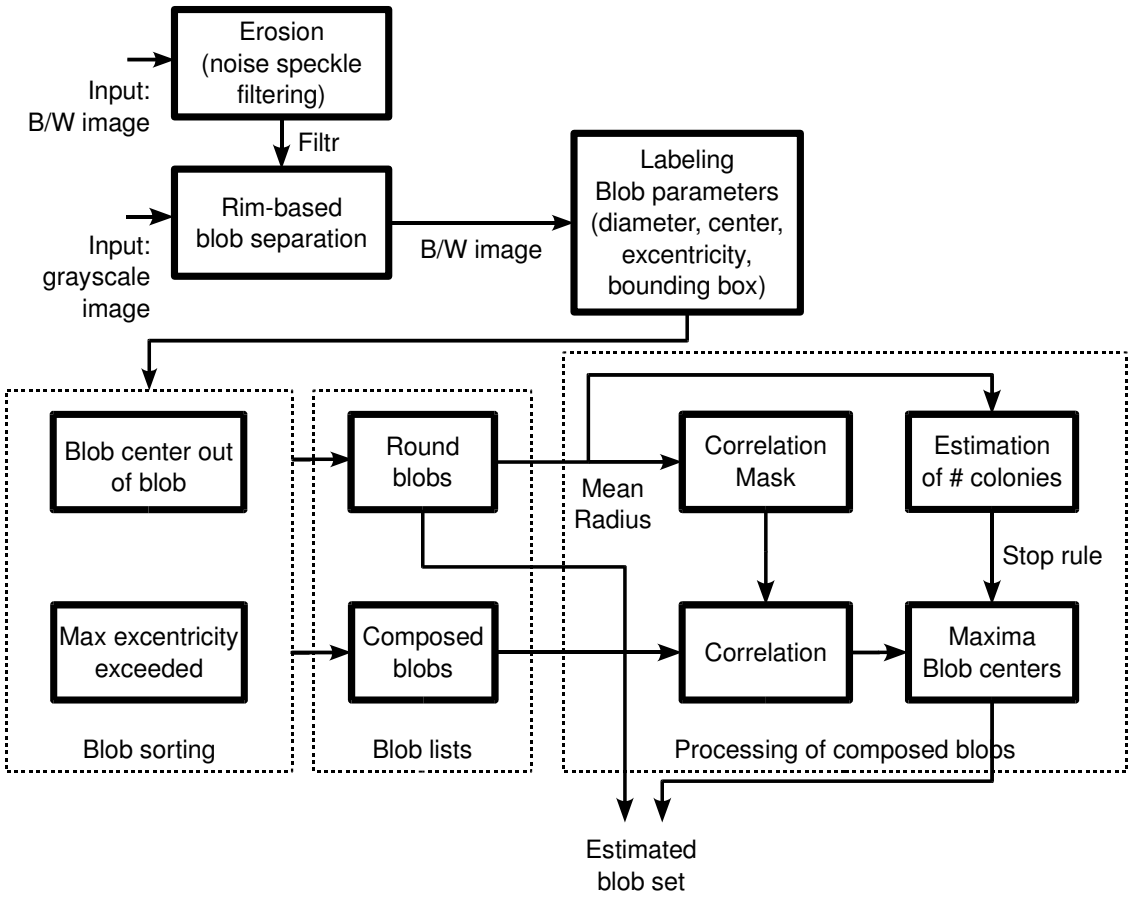


Figure 5: Processing flow of the DishPreprocessor() function

counted; after 5 repetitions (that is, the same number as used in the rim-based separation), 70 blobs were counted.

It should be emphasized that setting the threshold to the coordinate of histogram maximum is a fairly ad-hoc solution. More thorough evaluation is needed to judge the effectiveness of the method. A counterexample showing touching colonies that will not be separated using this rim-based method, is given in Figure 9.

4.3 Blob classification

The next step after the rim-based separation is to sort out the isolated colonies from the touching colonies. For all colonies, equivalent diameter, center of mass, bounding box and eccentricity are computed using function `regionprops()` from the Matlab Image Processing Toolbox.

The blob centers are first checked for location out-of-blob — that is, it is tested whether the blob center lies in the position of a blob “on” pixel or in the position of a background “off” pixel (see Figure 10 for illustration). In the later case, the blob is claimed a composed one.

Next, the blobs with eccentricity exceeding preset threshold are put to the list of composed blobs. This value of threshold is curently set to 0.8, based on test images. In future versions, this threshold should rather be a user-adjustable parameter.

4.4 Processing of composed blobs

The last step in counting is separation of the composed blobs. It is based on correlating circular mask with the contour of a composed blob. The radius of the mask is set to the mean radius of the blobs contained in the set of round blobs. The position of the individual colonies in the composed blob is then determined from the maxima of the correlation function, with the total number of colonies contained in the blob estimated from the size of the composed blob and the mean size of an individual colony. When this estimated number of colonies is reached, the search is stopped. The process is illustrated in Figure 11.

The correlation result is masked by the contours of the original blob and is further limited by a rectangular area of the width equal to the radius of the correlation mask, so that the maxima resulting from noise in the blob contour are suppressed as much as possible.

5 Experiments

The performance of the method has been tested on five test images. The results are summarized in Table 1 and in Figures 12– 16.

Table 1 shows, for each test image, a thumbnail of the image and the following data:

- the total number of colonies (after manual correction of the counting result).
- the number of the round colonies in the image, as detected by software (corresponds with the 'Round objects' in Figure 12– 16), also expressed as the percentage of the total number of colonies.
- Number of correctly and incorrectly detected colonies.
- Number of missed colonies.

Figures 12–16 show — for better insight — for each test image, the objects classified as round objects, the objects classified as composed objects, and finally the machine counting result (*before* any manual correction).

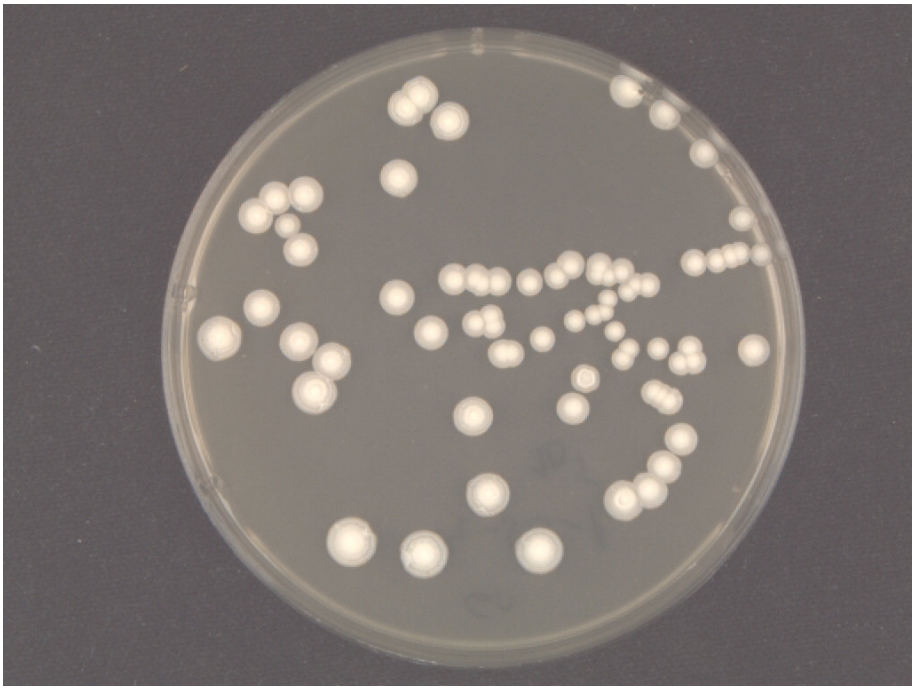


Figure 6: Sample image

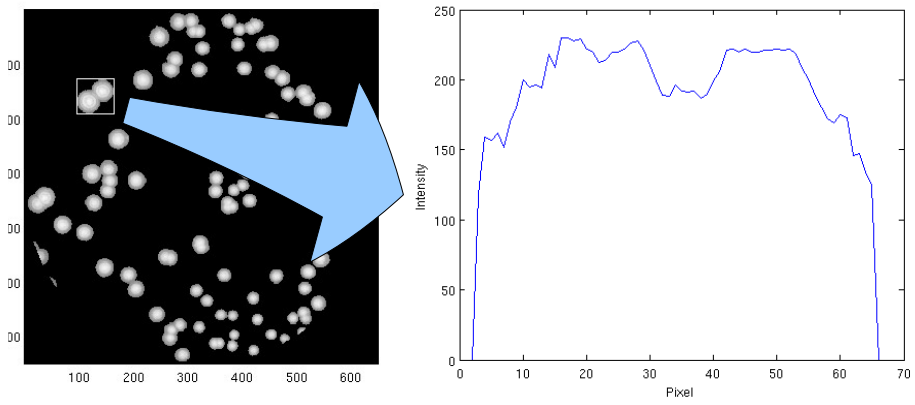


Figure 7: Composed blob, intensity values along its horizontal cross-section

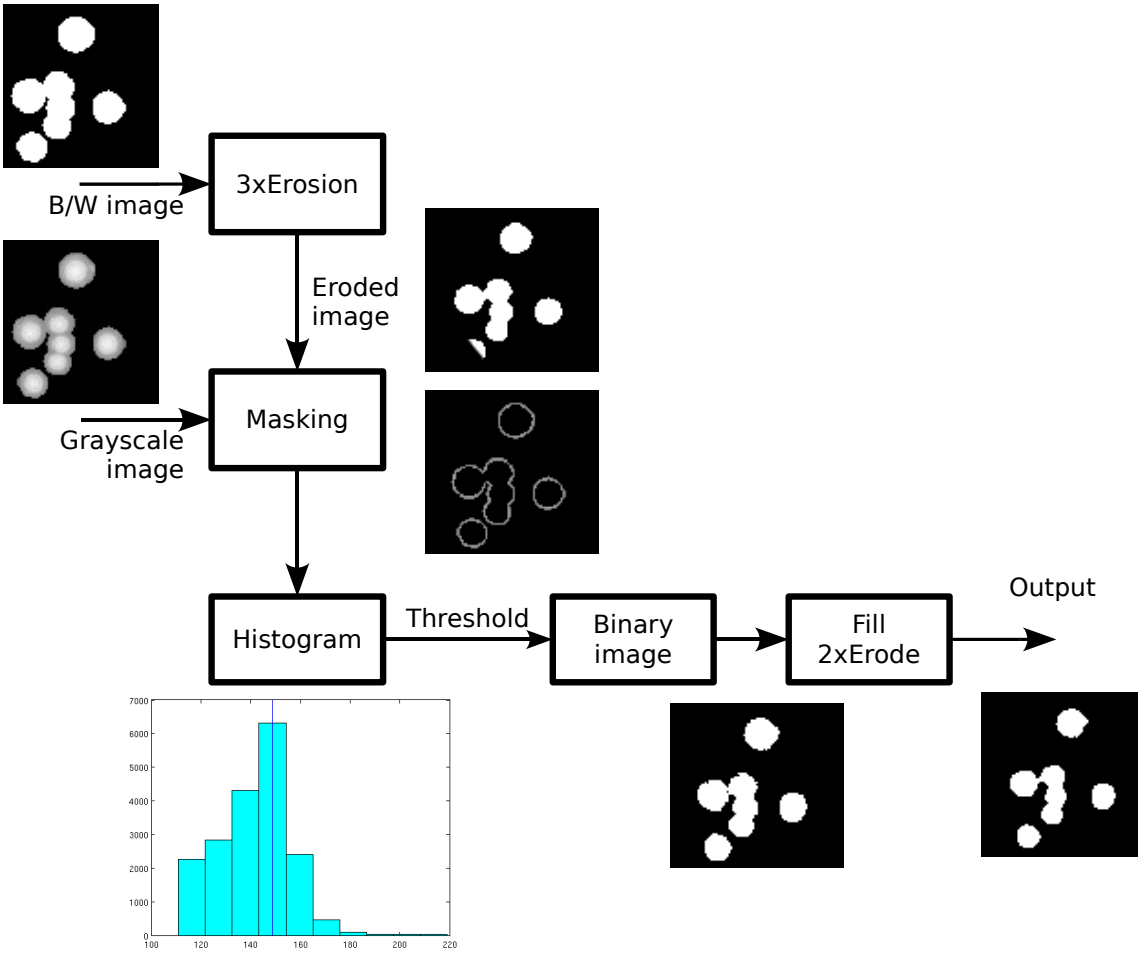


Figure 8: Rim-based blob separation

6 Concluding remarks

In this report, a demo for counting the yeast colonies in a Petri dish has been presented and its function explained in detail.

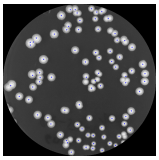
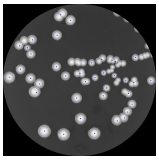
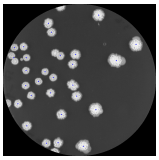
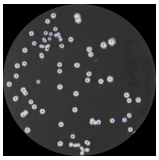
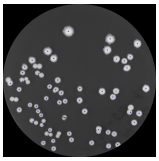
Image	Colonies Total	Round	Correct detection	Incorrect detection	Missed Colonies
Image 1					
	99	60 60.6%	79 79.8%	3 3%	20 20.2%
Image 2					
	66	28 42.4%	36 55%	3 4.5%	30 45%
Image 3					
	30	26 86.7%	27 90%	1 3.33%	3 10%
Image 4					
	73	48 65.8%	58 79.5%	1 1.4%	15 20.5%
Image 5					
	72	47 65.3%	55 76.4%	0 –	17 23.6%

Table 1: Results of colony counting

Five test images are included for evaluation of the performance of the method.

The method uses presorting of the colonies to round, isolated colonies and composed (touching) colonies, where the later are further processed to estimate the number of individual colonies in the composed blob. Figures, showing the results of the presorting to round and composed colonies are also included.

An average miss rate on the test image has been round 20%. It follows from Figures 12– 16 that the sorting to round and composed objects is set too lose. Also, the decomposition of the composed objects needs further improvement.

The counting definitely needs to try more classification methods and to perform more thorough performance analysis. These points, however, go beyond the scope of this report. Finally, it should be noted that already in this phase, the tool saves time and improves accuracy of experiment evaluation in the cooperating microbiology laboratory.

References

[1] J. Schier: Preprocessing of images of Petri dishes, Authorized software, ÚTIA AV ČR, Prague, 2009.

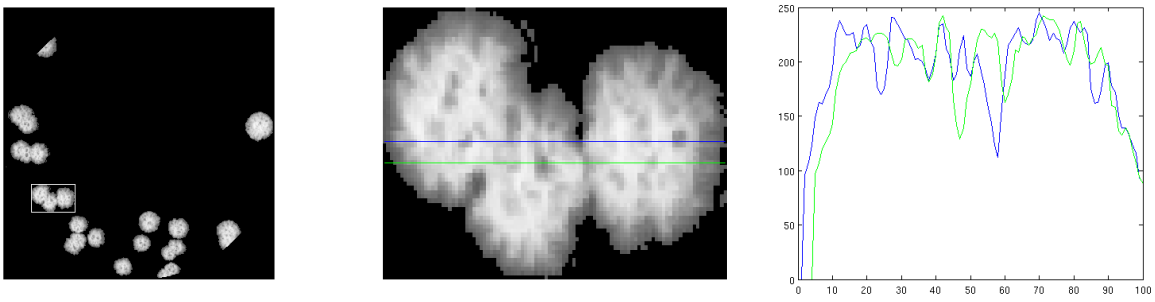


Figure 9: Touching colonies – counterexample for the rim-based separation

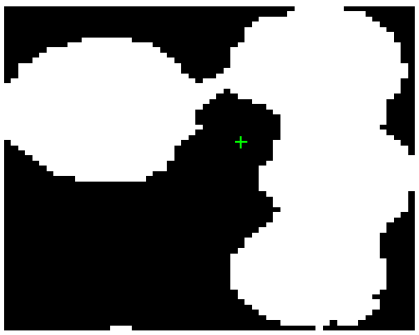


Figure 10: Illustration – center of mass out of blob

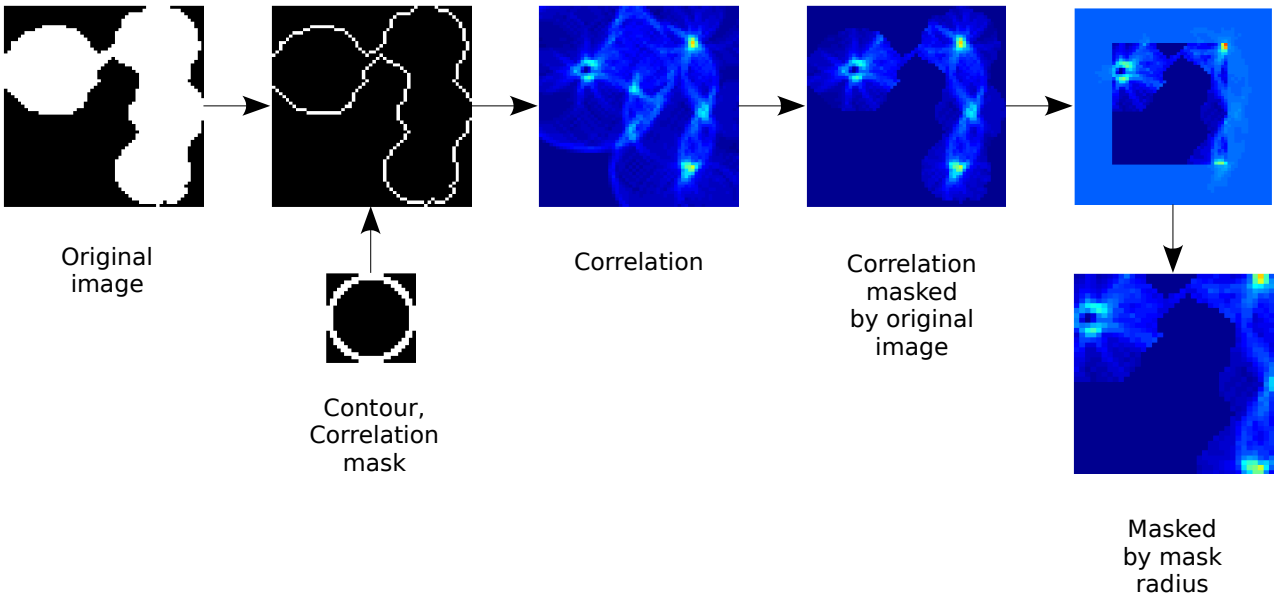
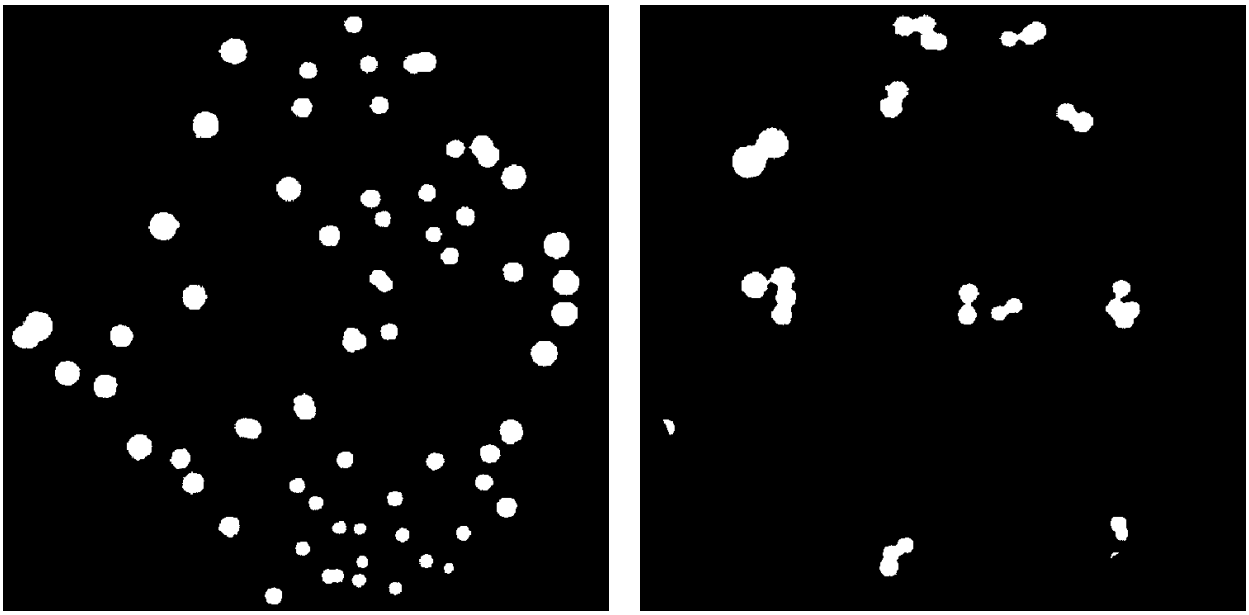
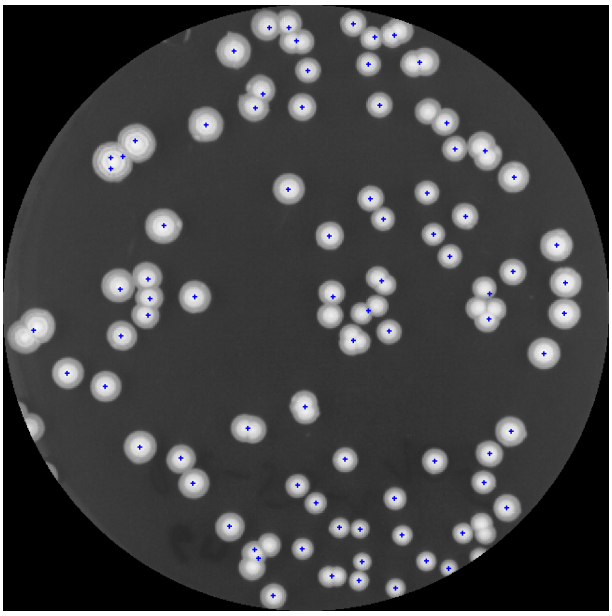


Figure 11: Blob separation flow

Image 1 — decomposition



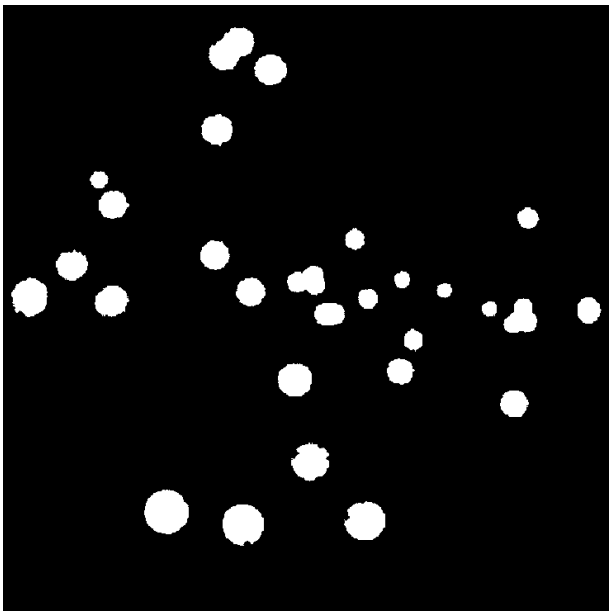
round objects composed objects



Counting result

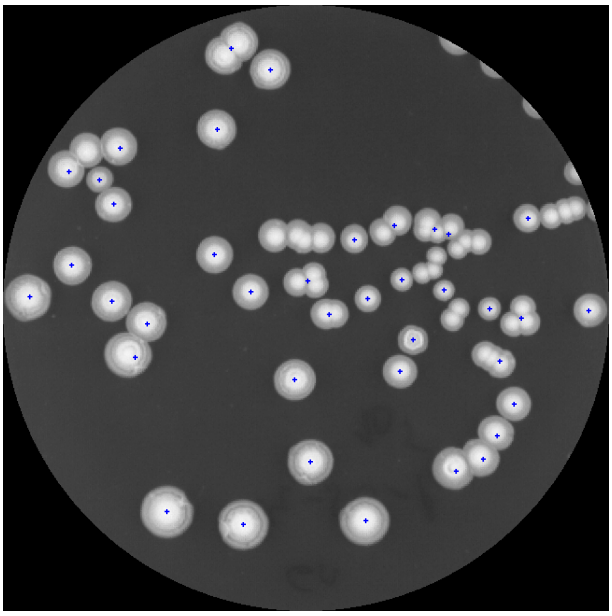
Figure 12: Test image 1: decomposition to round and composed objects

Image 2 — decomposition



round objects

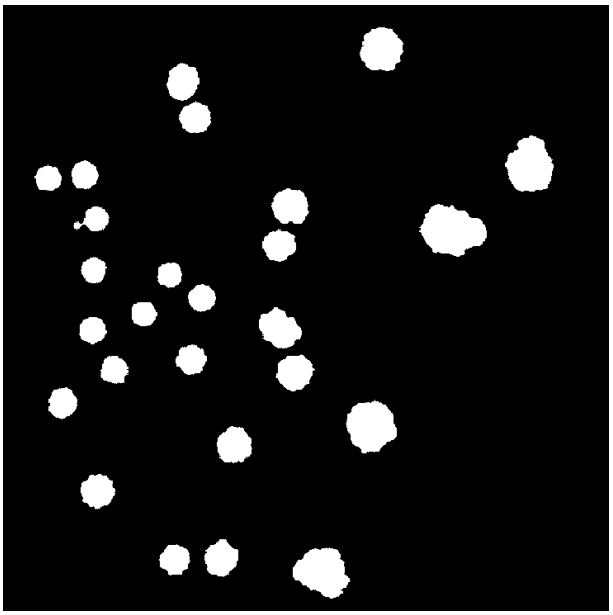
composed objects



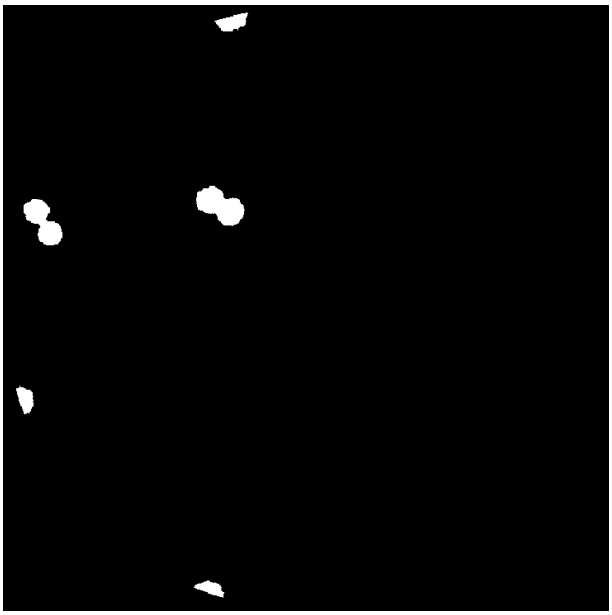
Counting result

Figure 13: Test image 2: decomposition to round and composed objects

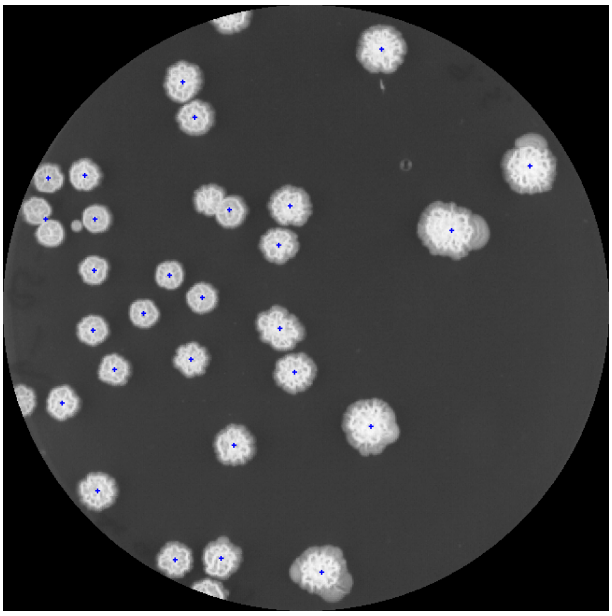
Image 3 — decomposition



round objects



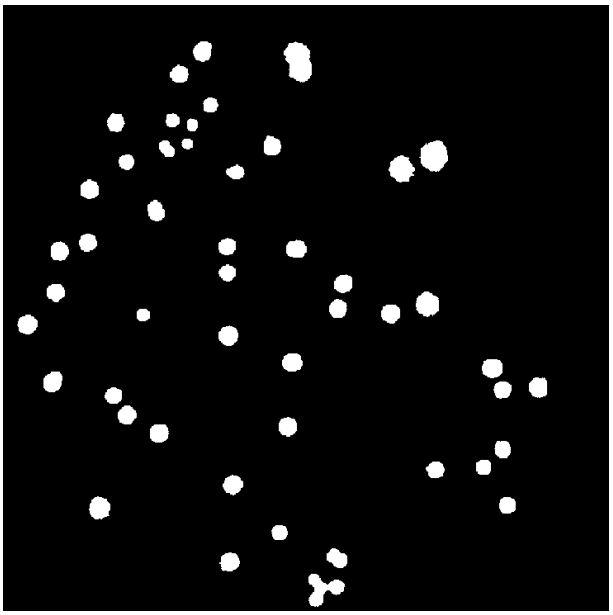
composed objects



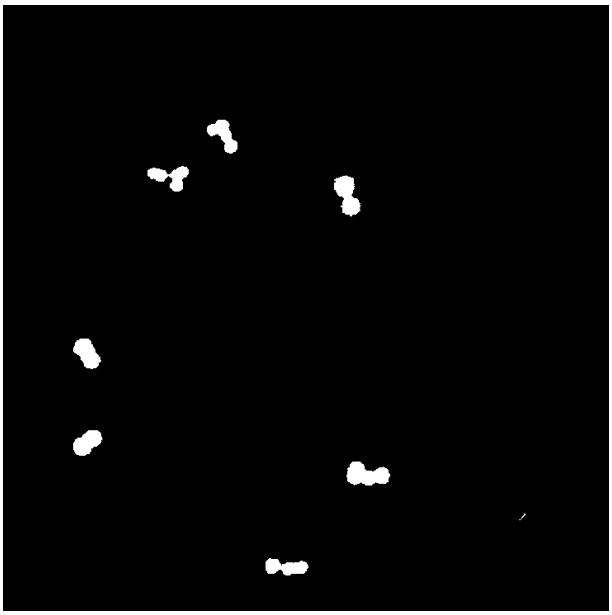
Counting result

Figure 14: Test image 3: decomposition to round and composed objects

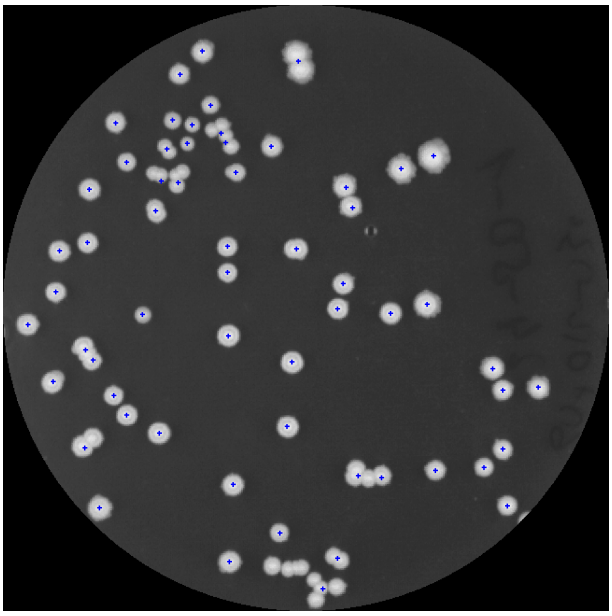
Image 4 — decomposition



round objects



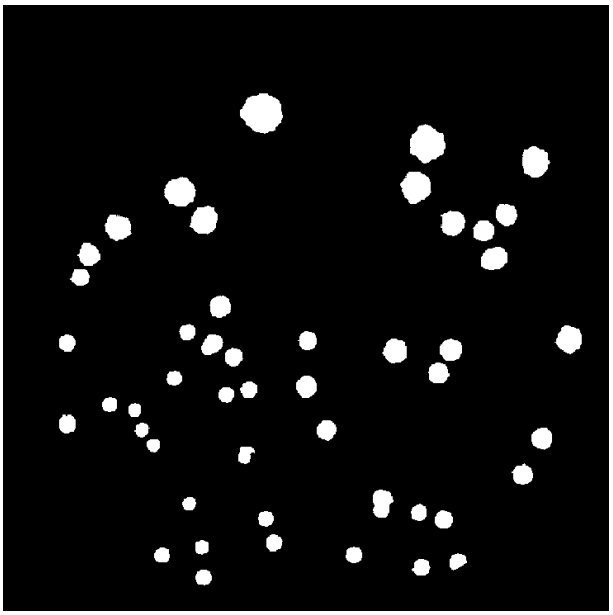
composed objects



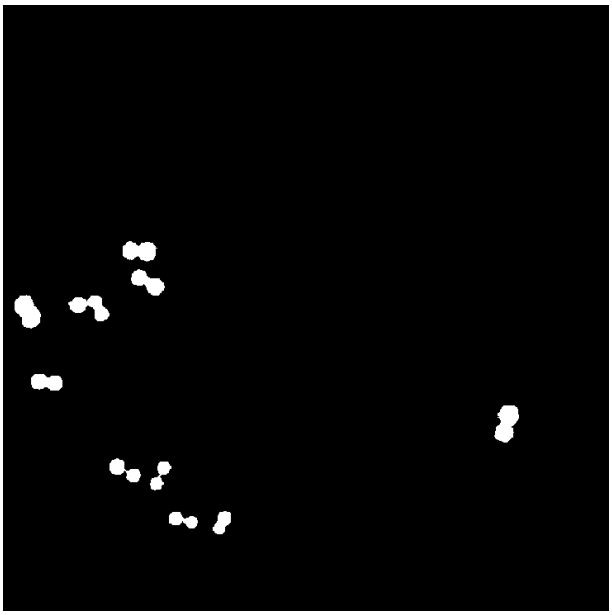
Counting result

Figure 15: Test image 4: decomposition to round and composed objects

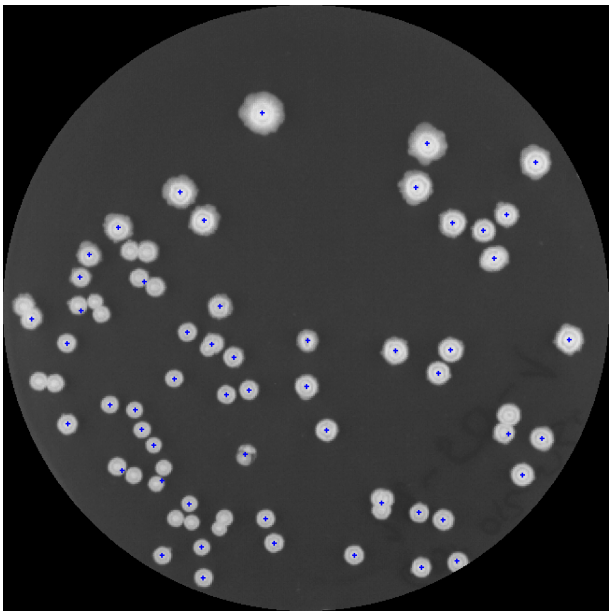
Image 5 — decomposition



round objects



composed objects



Counting result

Figure 16: Test image 5: decomposition to round and composed objects