

ELEN-0016 – Computer Vision Student projects 2023-2024

Prof. M. Van Droogenbroeck and R. Vandeghen

Version 1.0

Contents

1	Introduction	2
2	Tasks	3
2.1	Task 1: annotation	3
2.2	Task 2: development of a solution	3
2.2.1	Background subtraction	3
2.2.2	Counting	3
2.3	Task 3: evaluation	4
3	Organization and practical details	5

1 Introduction

The aim of the project is to “practice” computer vision with a practical application. To do so, your task is to implement a solution that *counts* and *detects* droplets and cells in high frame rate videos. The learning objectives are for you to

- handle images/videos,
- design methods,
- understand its components,
- evaluate the quality of the results, and
- be able to comment the solution.

In this project, you will work with series of frames (video) which display droplets containing a high number of cells (around 100). Droplets move inside of a tube, from left to right, with a number of 100 droplets per second, and a frame rate of 250 FPS. A sample frame is shown in Figure 1.

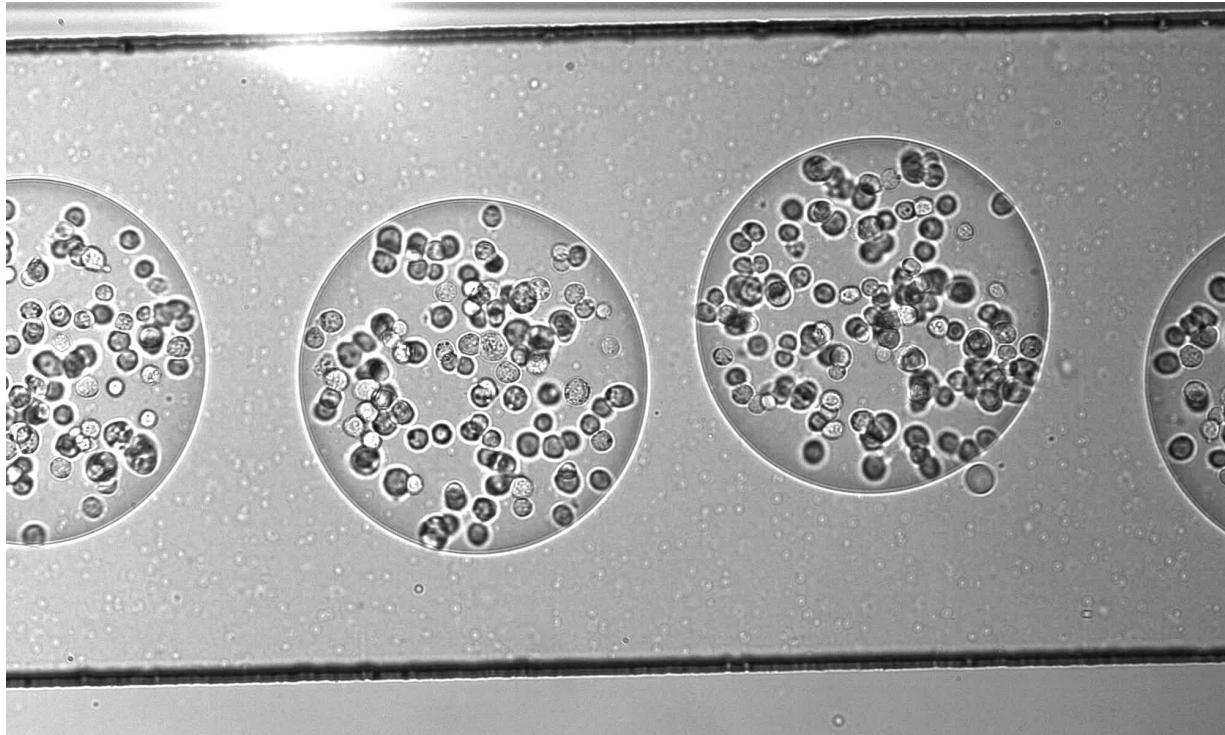


Figure 1: Example of a frame with droplets containing cells.

2 Tasks

To carry out this project, you will have to work on 3 main tasks:

- (Task 1) annotation of the data,
- (Task 2) development of a method, and
- (Task 3) evaluation of your solution. These tasks are discussed hereafter.

2.1 Task 1: annotation

For this task, you are requested to annotate 1 sequence per person. Annotations are markers for cells present inside each droplet, as well as the bounding boxes of the droplets. You are asked to annotate the markers for **30 consecutive frames, and the bounding boxes for 50 consecutive frames**. Please follow the instruction given during the presentation. At the end of the annotation process, the sequences of each person will be shared, so that you can use at your convenience all the data. The annotation process must be finished and sent by **Friday, the 6th of October**.

The annotations must be gathered inside a json file, following the structure presented in the file *example.json*. Note that the top left of the frame is the origin.

2.2 Task 2: development of a solution

2.2.1 Background subtraction

The first task that you have to implement is a background subtraction method. For this part, you will need to implement your solution by your own (you cannot use an open-source code). If your implementation is based on a known algorithm (MoG, ViBe, ...), you will need to specify and reference the original method. For the evaluation part, you need to provide a qualitative assessment and explain how you could proceed if you had to make it objective. This first subpart of task 2 is due by **Friday, the 3rd of November**. You are asked to send a report of **maximum 2 pages** and your source code.

2.2.2 Counting

The main task that you have to undertake is the counting the number of cells per droplet. To do so, you are allowed to use the method(s) that you want, as far as you understand all its components. You will have to discuss the choice of your method regarding the performance/inference time trade-off.

By the end of the project, you have to provide the code(s) of your solution. You are allowed to use the programming language and the libraries that you want. However,

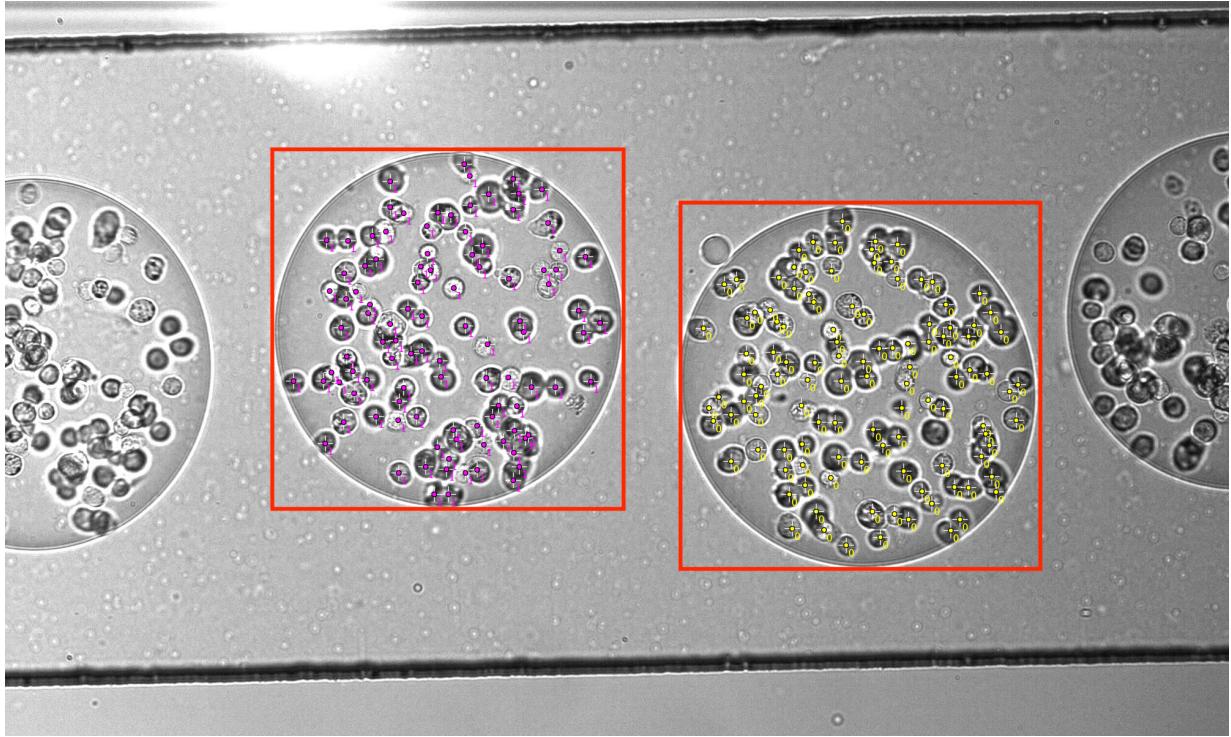


Figure 2: An annotated frame with the bounding boxes and markers draw on the image.

we will provide helps only with Python. Also note that, to avoid any issue related to plagiarism, you need to explicitly provide the reference of any code that is not entirely yours. Note that taking some lines of existing code is also considered to be plagiarism if you do not provide a link to the original version properly.

2.3 Task 3: evaluation

Finally, you have to assess the performance of your method. To do so, you are asked to

- provide a quantitative assessment of your method, by comparing the predictions with respect to the ground truths.
- propose the comparison metrics that you consider the most suitable for your application.
- understand those metrics.

A validation and a private test set will be available to evaluate your method.

3 Organization and practical details

1. Groups comprise 3 persons (no exception).
2. The final presentation will be an oral presentation with a poster and a live demo.
3. Programming languages: C/C++ or Python are allowed. In the code, it is mandatory to clearly identify (with start and end tags) code extracts taken from others programmers. Any failure to identify the extracts is considered as plagiarism and it will result in a global note of 0/20 for all the members of the group!