Droplet Detection and Cell Counting: A Computer Vision Approach

Brandoit Julien, Gerard Manon, de Thibault Adrien
University of Liège
Project year: 2023 - 2024

Abstract

This project presents a resilient computer vision solution designed for the simultaneous detection and counting of droplets and cells, with a distinctive feature of intentionally excluding dependencies on machine learning. Notably, the system addresses the unique challenge of identifying cells encapsulated within flowing droplets as they continuously traverse through a channel. Unlike methods requiring extensive datasets, our approach prioritizes ease of implementation in various scenarios.

Utilizing computer vision techniques such as the Hough transform and ellipse fitting for contour detection, and background subtraction based on a learned background model, the system incorporates a tracking component for detailed statistical analysis specific to individual droplets. The method's adaptability extends to dynamic scenarios, including instances of droplet contact. The background subtraction method and ellipse fitting isolate moving droplets, and the Hough transform ensure cell counting within each droplet. Tracking is incorporate using a statistical model of motion.

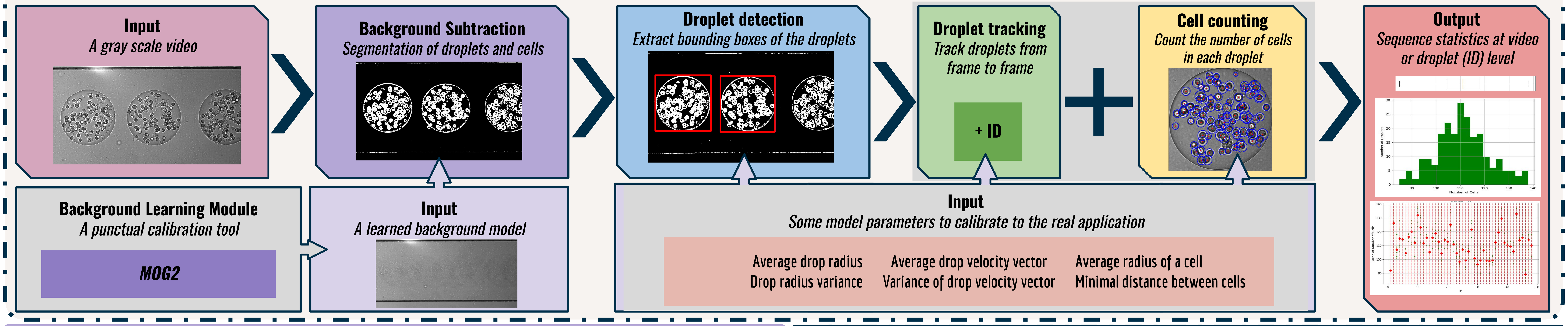
Acknowledging that the absence of machine learning compromises the algorithm's speed compared to ML-based counterparts, this tradeoff is made deliberately to enhance versatility. Quantitative evaluation demonstrates the efficacy of our approach. The methodology emphasizes efficiency, offering a rapidly implementable and reliable solution for droplet and cell analysis.

This project contributes a valuable approach to droplet and cell analysis, suitable for scenarios where machine learning datasets are limited or unavailable. The deliberate compromise in speed ensures broad applicability across diverse experimental setups.

Project Objectives

- **Background Subtraction Method:**
  - Develop a background subtraction method based on a learned background model to accurately isolate moving droplets from the background noise.
  - Implement a mechanism to facilitate the learning of the background model, allowing recalibration of the system when necessary.
- **Droplet Detection and Individual Cropping:**
  - Design an algorithm capable of detecting moving droplets even in scenarios of contact between droplets and ensure the individual cropping of each droplet from the image.
- **Consistent Droplet Tracking:**
  - Implement a tracking system to assign a consistent identifier (ID) to each droplet from one frame to another, enabling the tracking of individual droplets throughout the sequence.
- **Cell Counting within Cropped Droplets:**
  - Develop a method to count cells within each cropped droplet, providing detailed insights into the cellular content of individual droplets.
- **Parallelization for Speed Improvement:**
  - Integrate parallelization techniques into the codebase to enhance the speed of the solution, providing a foundation for future improvements in computational efficiency.

Global Pipeline of The Method



Background Subtraction

Background suppression is based on a binary threshold applied to the difference between the processed frame and the background model given in input. A median filter is applied before the threshold to remove *salt-and-pepper noise* and make the method more robust. The background removal step can be easily parallelized to process several frames at once.

The method can be calibrated periodically to take into account the natural evolution of the background in the working environment. A background learning script is provided to generate the background model. This learning is based on a MOG2 type method (based on the *OpenCV* implementation) on part of a sequence.

Background subtraction and learning are performed on smaller versions of the images to increase execution speed without any visible detriment to the accuracy of the rest of the pipeline.

Droplet detection

Droplet detection is carried out on a reduced version of the images to increase speed. This step can also be easily parallelized.

External contours are extracted from the image using the simple chain approximation. Their convex hulls are then determined and are filtered. Only those whose perimeter does not exceed a threshold value - based on the average radius of a drop and its standard deviation - are retained. This filtering is necessary to discriminate the contours of cells and cell aggregates from those of bubbles. A final filter is carried out to detect hulls whose perimeter is too large - using another threshold value based on average and deviation - indicating that bubbles are in contact and need to be separated.

Separating touching bubbles
Touching drops are separated and added separately to individual hulls. The touching drops are extracted using the group hull mask, then the original contour is filled in to create a solid version of the touching drops. An erosion with a circle of radius slightly smaller than the average radius is then applied, separating the drops. The connected components of the result are recovered and processed separately. For each of these, a dilation using the same kernel as for the erosion is applied. The result is an approximate version of the separated drops. The hulls of these approximations can then be added to the list of individual hulls.
Treatment of individual droplet hulls
The set of hulls that have passed the various filters are considered to be those of individual drops. An ellipse is fitted to the contour to minimize the algebraic distance. Finally, the axis-aligned bounding boxes of those ellipses are calculated and kept if they do not go outside the image.

Droplet tracking

Drop tracking aims to assign a unique ID to each drop consistently across frames by predicting its future position based on the physics of the environment (mean and variance of the velocity).

$$\vec{x}_{ID,f+1} \sim \vec{x}_{ID,f} + \mathcal{N}(\mu_v; \sigma_v^2)$$

The method identifies the most probable bounding box in each frame using the probability density of the future position prediction. If the prediction suggests the drop has left the screen, it is no longer assigned in subsequent frames.

The method is resilient to false-negative bounding box detection by applying a threshold on the probability density. If the highest probability bounding box falls below this threshold, the ID is not assigned, but the theoretical position is retained for future frame assignments.

Cell counting

In the process of foreground extraction from the video, cells are detected using a downsized version of the frames and parallelization techniques for improved computational efficiency.

Extracting the foreground
The initial step involves isolating droplets from the frames to optimize runtime by focusing on the anticipated cell areas. To address gaps in the cells, a closing operation is applied. Subsequently, a dilation is performed on the Background Subtraction result. The resulting mask is then combined with the actual frame using a bitwise *'and'* function. Finally, to reduce noise, a bilateral filter is applied, resulting in a version of the frame without the background.
Cells detection
For cell detection, the Circle Hough transform from *OpenCV* is employed. The upper threshold for edge detection and the threshold for center detection are set empirically at 50 and 18, respectively.

Results

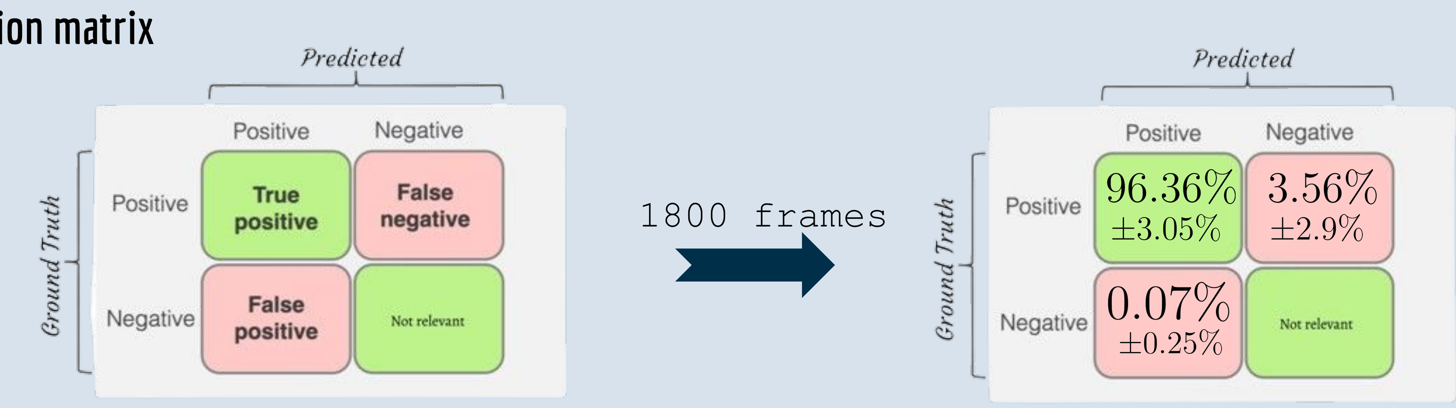
Test sample
We were able to evaluate the performance of our method using an annotations dataset made by hand used as an approximation of a ground truth. It was statistically cleaned by removing outliers. The background model was calibrated for each sequence.

Composition of the test set : 36 videos

Definition of the accuracy metrics used and results

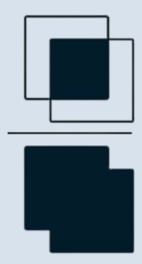
Droplets detection

Confusion matrix



Jaccard Index

Jaccard Index =



4336 true-positives bounding boxes

$$\text{Jaccard Index} = 97\% \pm 0.79\%$$

Droplets tracking

Accuracy of assigned IDs

$$\text{accuracy} = \frac{\#\text{ boxes with correct ID}}{\#\text{ boxes}}$$

4336 true-positives bounding boxes

$$\text{accuracy} = 99\% \pm 1.12\%$$

Cell counting

Difference between prediction and ground truth

$$\text{difference} = \frac{\#\text{predicted cells} - \#\text{real cells}}{\#\text{real cells}}$$

2600 true-positives bounding boxes

$$\text{difference} = 1.1\% \pm 9.9\%$$

Distance of prediction compared to random

$$\text{performance} = \frac{d(\text{random}, \text{real})}{d(\text{predicted}, \text{real})}$$

2600 true-positives bounding boxes

$$\text{performance} = 3.2 \pm 0.55$$

$$d(P,T) = \frac{1}{2|P|}\sum_{\mathbf{p}\in P}\min_{\mathbf{t}\in T}\left(\|\mathbf{p}-\mathbf{t}\|^2\right) + \frac{1}{2|T|}\sum_{\mathbf{t}\in T}\min_{\mathbf{p}\in P}\left(\|\mathbf{p}-\mathbf{t}\|^2\right)$$

Output : 25 FPS (average) [AMD Ryzen 5 Radeon, 2.8GHz]

Conclusion

Our method excels in bubble detection with minimal false positives and few false negatives. The bounding boxes effectively surround bubbles, and tracking performances are excellent. Despite the cell count having a mean close to the theoretical mean, the observed variance is relatively high, potentially causing issues in certain applications. It's important to note that this result could come from imperfect annotations nonetheless the predicted positions of cells are significantly better than random.

We have successfully achieved all stated objectives, affirming the robustness and practical applicability of our integrated pipeline for droplet and cell analysis. Our method boasts commendable accuracy, offering a rapid and easily implementable solution without requiring extensive annotated data. While our approach excels in versatility and accuracy, it may not match the speed of machine learning (ML)-based alternatives. An advantageous use for our method could be as a transitional tool in new environments, helping build necessary annotations for an ML method that would subsequently deliver faster performance in practice.