ELEN0016-2 Computer Vision Project: Cell Detection

Lambermont Romain, Louis Arthur, Navez Tom

LIÈGE université

lambi702/CV-Video_annotation

Introduction

Our computer vision project unfolds in two key phases:

- **Background Removal:** Meticulous elimination of backgrounds from original images, a crucial prerequisite.
- **Droplet and Cell Detection:** Our central focus, navigating from background elimination to the presentation of our refined solution.

In this presentation, we invite you to explore our methodology, highlighting the challenges and successes encountered during the enhancement of computer vision capabilities in the microscopic domain.
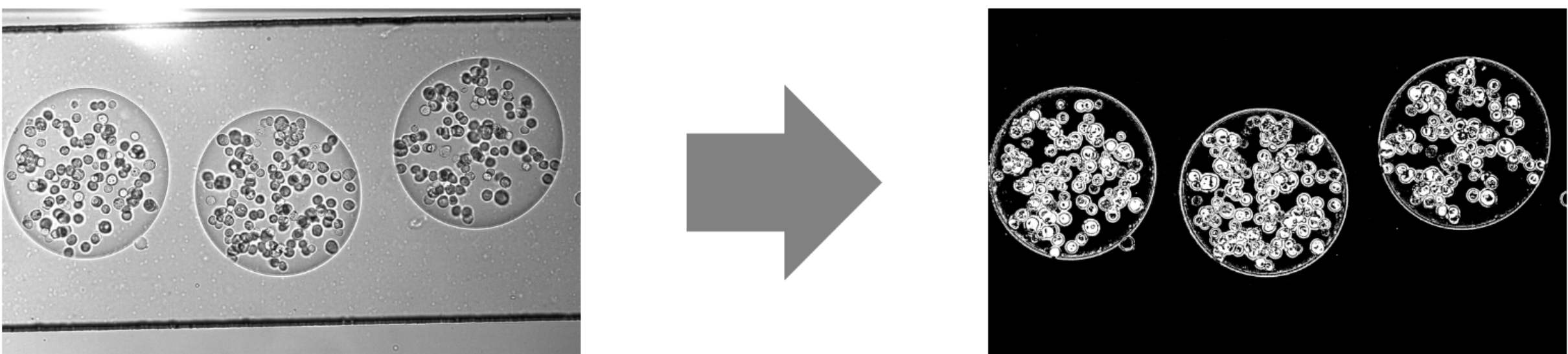
Creating the Mask

Figure 1. Creation of the mask of an example frame.

Mask importance

Flawless execution is paramount because any imperfections at this stage introduce noise that persists throughout subsequent steps. Consequently, considerable effort is invested in refining our masking method, extensively testing it across diverse sequences. Addressing exposure variations emerged as a significant challenge in this process.

Initially, we explored an energy score-based method for background reconstruction and the ViBe algorithm [1], but it yielded unsatisfactory results. We then adopted a simpler approach: computing the mean of pixels across 100 frames to establish the background. Subtracting this background from each frame and applying a threshold allowed us to rapidly and efficiently generate the desired mask.
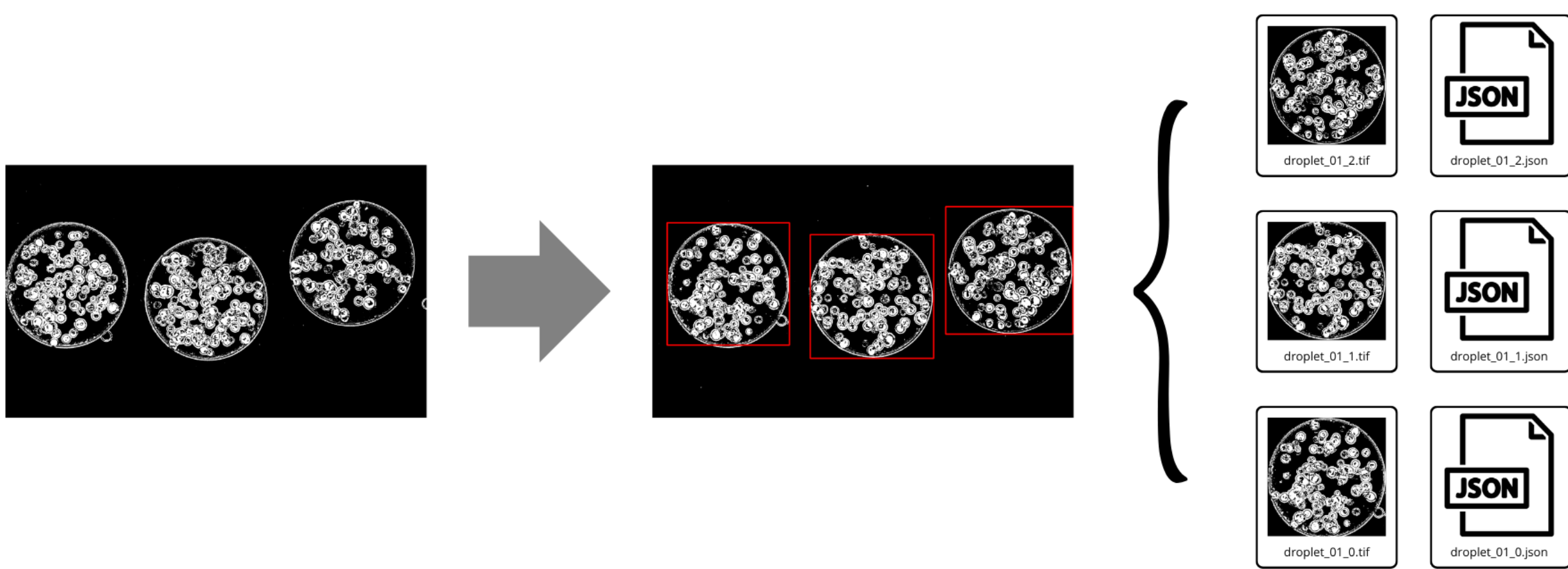
Isolating the Droplets

Figure 2. Computation of bounding boxes and creation of `.tif` and `.json` files for each box.

Bounding boxes were derived from the mask using the Circular Hough Transform

function from OpenCV [2]. We fine-tuned the function parameters to selectively detect circles within a specific radius range. Subsequently, based on the circle's center and radius, we computed a square around it. Addressing ellipsoidal cells posed a challenge, requiring additional parameter tuning in the function.
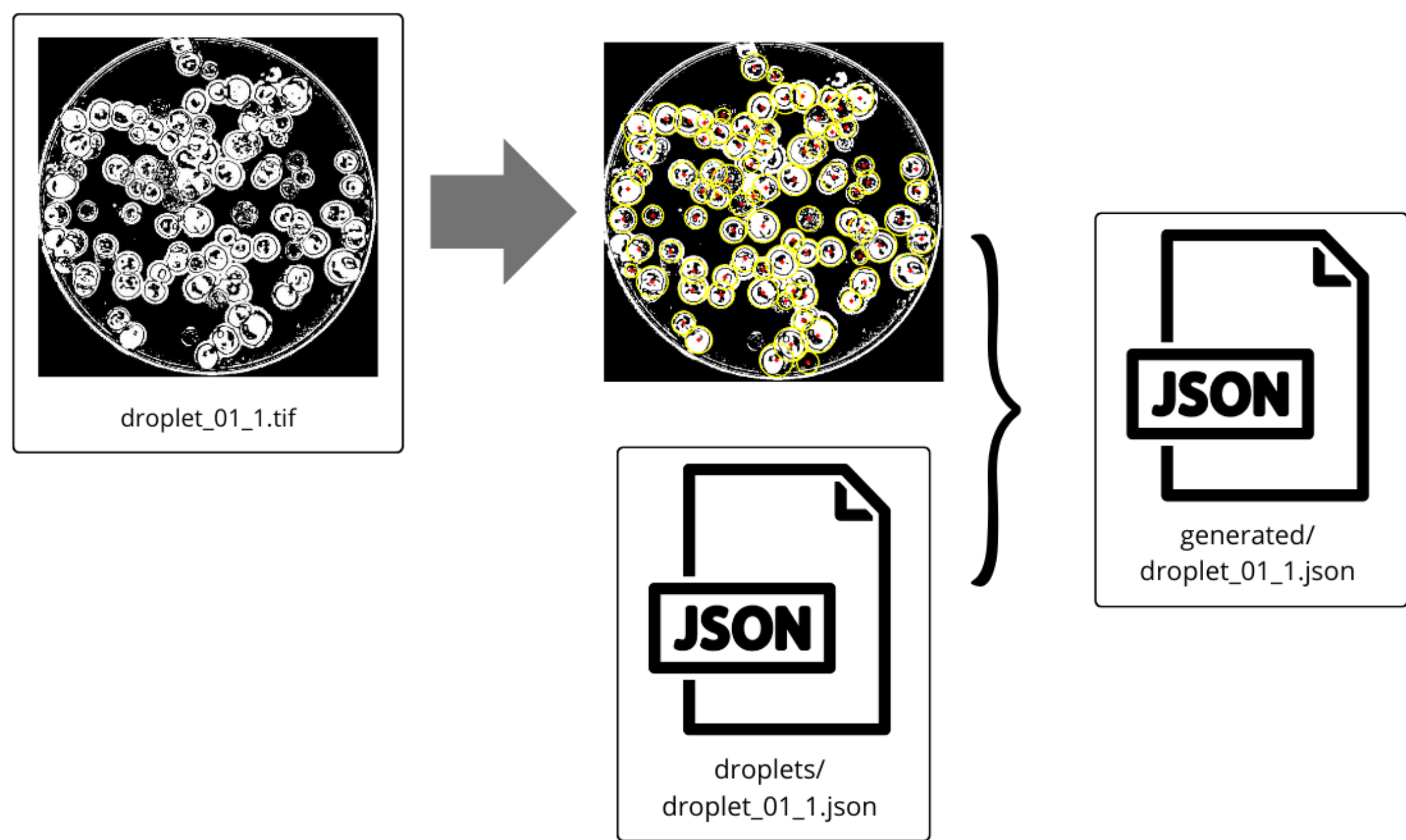
Counting and Locating the Cells

Figure 3. Counting and locating the cells inside a zoomed-in droplet and update of the `.json` file.

The method used is the same as before to detect the droplet but on isolated droplet and to a smaller scale.

Functions used from CV2 [2]

- `findContours`:
  - Used between the steps of mask creation and the identification of the bounding boxes.
- `HoughCircles`:
  - Task 1: Used to identify the droplets, allowing us to create the bounding boxes.
  - Task 2: Used within isolated droplets to pinpoint cells.
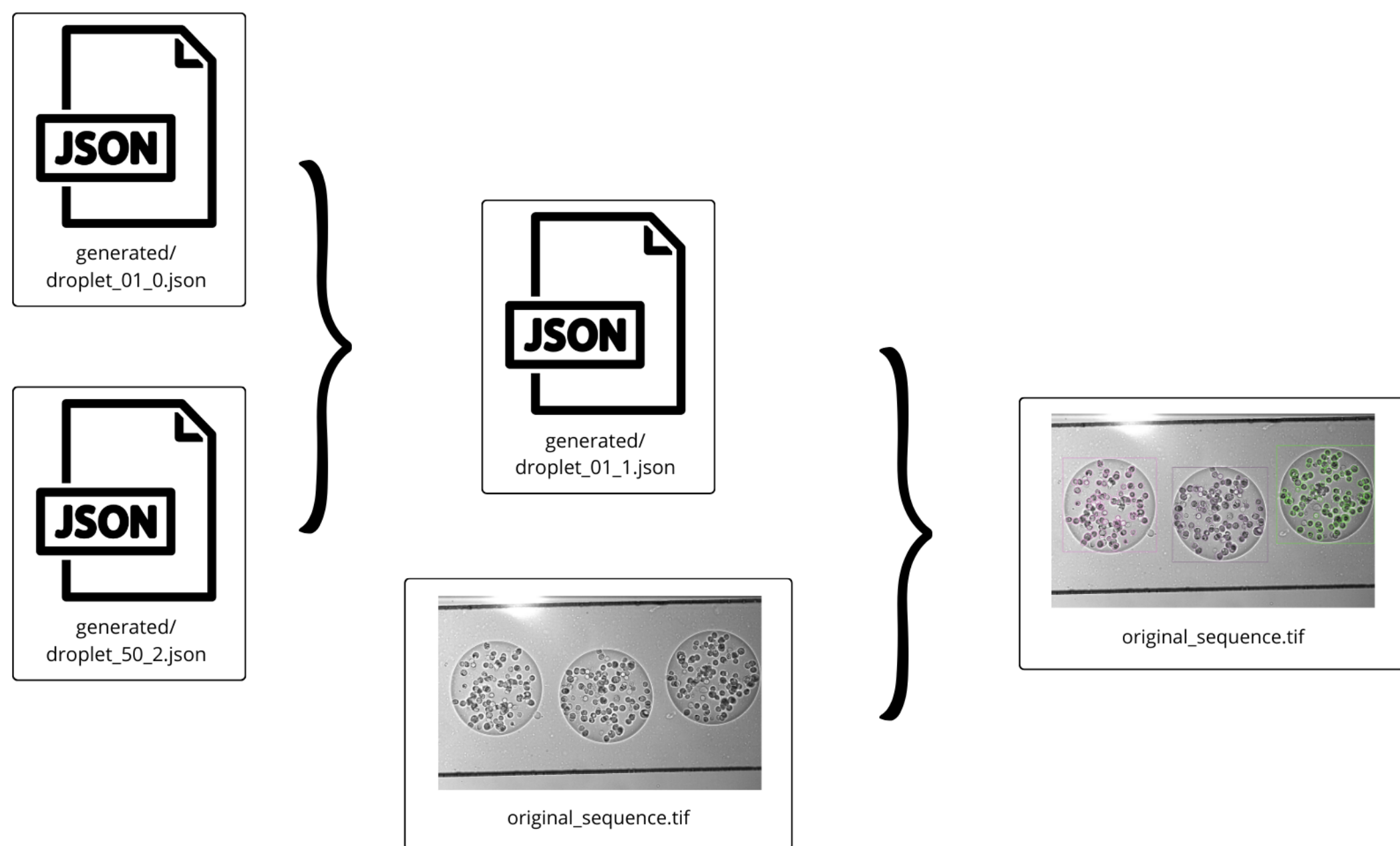
Combining Data and Sequence Annotation

Figure 4. Combination of the different `.json` files and annotation of the original sequence.

Statistics

	Mean	std. dev.
Mask Creation	1.66 s	±35.6 ms
BB Computation	6.38 s	±145 ms
Counting Cells	2 s	±64.8 ms
Sequence Annotation	50.8 ms	±2.81 ms

Table 1. Elapsed time computed over 7 iterations for 30 frames each.

Alternative Methods and Discussion

We implemented a Convolutional Neural Network (CNN) on resized, isolated droplets (448×448 pixels). Two convolution operations were performed, followed by flattening into a standard fully convolutional network. The flattened image was then concatenated with bounding box coordinates to convey droplet position within the full image.

For loss computation, a blend of Earth Mover's Distance (EMD [3]) and Mean Squared Error (MSE) for cell count was employed. Regrettably, this approach faced challenges and identified issues:

1. Loss sensitivity to annotation quality.
2. Network simplicity as a potential limitation.

3. Complexity of Earth Mover's Distance hindering correct usage.
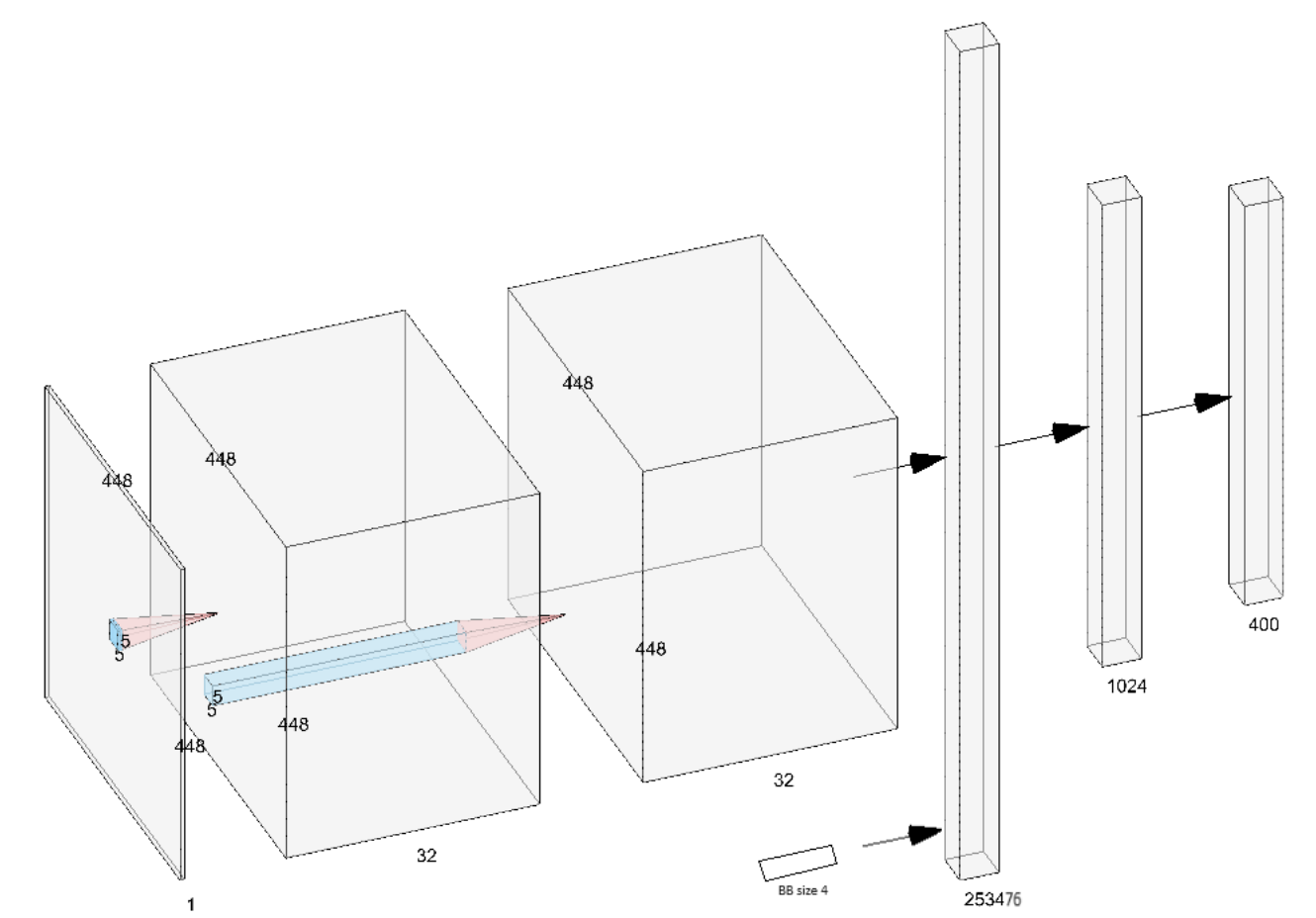

Figure 5. Neural Network tried for the project.

The neural network could have been a better idea if our purpose was only to count the number of cells inside the droplets, but since we also wanted to locate them, we decided to move on and go back to non Deep-Learning based Computer Vision.

References

[1] Marc Van Droogenbroeck and Olivier Barnich. **"ViBe: A Disruptive Method for Background Subtraction". English. In: Background Modeling and Foreground Detection for Video Surveillance. Chapman and Hall/CRC, July 2014. DOI:** 10.1201/b17223.

[2] G. Bradski. **"The OpenCV Library". In: Dr. Dobb's Journal of Software Tools (2000).**

[3] Jean Feydy et al. **"Interpolating between Optimal Transport and MMD using Sinkhorn Divergences". In: The 22nd International Conference on Artificial Intelligence and Statistics. 2019, pp. 2681–2690.**

[4] F. Rozet. **In: Poster layout. URL:** https://github.com/francois-rozet/sleek-poster/tree/master.