IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 May 2024, accepted 19 June 2024, date of publication 1 July 2024, date of current version 12 July 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3421637*

RESEARCH ARTICLE

# Embedded Blur-Free Single-Image Acquisition Pipeline for Droplet Microfluidic Imaging Flow Cytometry (IFC)

**FARIHA AFRIN[1], (Member, IEEE), KAISER PÄRNAMETS[1], (Member, IEEE), YANNICK LE MOULLEC[1], (Senior Member, IEEE), ANDRES UDAL[2], (Member, IEEE), ANTS KOEL[1], (Member, IEEE), TAMAS PARDY[1,3], (Member, IEEE), AND TOOMAS RANG[1,3], (Senior Member, IEEE)**

[1]Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, 19086 Tallinn, Estonia
[2]Department of Software Science, Tallinn University of Technology, 19086 Tallinn, Estonia
[3]Department of Chemistry and Biotechnology, Tallinn University of Technology, 19086 Tallinn, Estonia

Corresponding author: Fariha Afrin (fariha.afrin@taltech.ee)

This work was supported in part by Eesti Teadusagentuur (ETAg) under Grant PRG620, in part by Estonian Research Council project under Grant PSG897, and in part by Estonian Education and Youth Board Project Artificial Intelligent Internet of Things (AIoT)∗5G under Grant ÕÜF11.

**ABSTRACT** Good quality of single droplet image acquisition in imaging flow cytometry (IFC) is crucial for a wide range of biological analyses. Recently, there have been significant advances in droplet microfluidic data analysis; however, acquiring blur-free single object images is still a great challenge because of the tradeoff between high flow rate and hardware setup complexity and cost. State-of-the-art hardware setups for blur-free single image acquisition are often complex, cumbersome, and not portable, limiting their suitability for point-of-care diagnostics. Moreover, motion blur and duplicate droplet image acquisition can occur with flow rate variation. To address these issues, this paper proposes a lightweight imaging pipeline for acquiring blur-free single droplet images for portable applications; this pipeline is capable of acquiring every single droplet image. While most of the existing literature focuses on complex hardware setups, utilizing high frame rate cameras that are not cost effective and complex optical solutions for droplet focusing, our pipeline utilizes minimum hardware and a lightweight algorithm for detecting, counting, and acquiring single object images from the video stream. The proposed pipeline was evaluated experimentally using videos of fast-moving droplets in which the input fluid flow rate was as high as 67.7 $\mu$L/min. The proposed pipeline achieves 100% counting accuracy on the tested videos and 2 ms, 25 ms and 10 ms processing time for each droplet on a desktop PC, single-board computer Raspberry Pi-4, and Nvidia Jetson Nano, respectively. This yields a maximum of 500, 40, and 100 blur free detected droplets per second (DPS), respectively. The Jetson Nano implementation, achieving 100 DPS with processing time of 10 ms, is faster than existing similar studies and fast enough for the target application. The results suggest that the proposed lightweight pipeline is suitable for efficient single object image acquisition in IFC on an embedded portable platform.

**INDEX TERMS** Droplet, image acquisition, imaging flow cytometry, microfluidic, single-board computer.

## I. INTRODUCTION

Imaging Flow Cytometry (IFC) is a combination of optical image acquisition, often involving microscopy, and flow cytometry; IFC is a key enabling technology in the drug discovery [1], [2] and medical diagnosis fields [3], [4]. IFC enables diverse morphology analyses, which are performed on a single-object image using a fully automated or partially automated system in laboratory or point of care (POC) devices [5]. Hence, a high-quality single-object image is indispensable for the accurate analysis of the specific object, as well as for machine learning/deep learning training [6], [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Mahmoud Al Ahmad.

The analytical results are used in several biomedical applications [8], such as early disease detection and circulating tumor cell detection [9], [10].

However, the process of acquiring a blur-free single-object image is challenging in high throughput IFC, especially if the camera settings are not synchronized with the flow of the object. For example, motion blur can happen if the object flow is too fast relative to the camera exposure time, and noisy images can be captured if the camera exposure time is too long [11], [12].

Some existing studies have focused on developing a dedicated optical system for acquiring single-object images at a high throughput and speed which makes the whole system bulky and expensive [13], [14]. Other examples such as the commercial IFC system "ImageStream" [15] as well as some other IFC systems, achieve cell focusing by using a precise pump to control the fluid flow rate and a velocity detection subsystem to control the time-delay integration readout [1]. Parallelized microchannels are also used to decrease the flow speed for a specific throughput and to obtain a single bright field object image [16], [17].

Moreover, the optical time-stretch imaging method, which consists of a complex optical setup, enables single-image acquisition and exploits spatial and temporal dispersion [18], [19]. In [20], the particle flow, controlled for fixed speed and motion, is frozen by the ultrafast shutter speed of the camera to obtain blur-free images. In [21], an optomechanical virtual motion freezing fluorescence imaging (VIFFI) method was developed for high throughput (>10,000 cells/s) imaging. However, it has a complex hardware setup that consists of an excitation beam scanner, a speed-controlled polygon scanner, and a series of timing control circuits in order to increase the exposure time of the image sensor.

The integration of these additional hardware systems for controlling the flow increases the complexity and decreases the cost efficiency. Image reconstruction can be applied without integrating additional hardware, yet this requires blur identification and data preprocessing [22].

Other studies focused on applying machine learning or deep learning algorithms to acquire and then analyse single-droplet images. A demonstration of TensorFlow feasibility in classifying red blood is presented in [23]. AI assisted pathogen detection with 99.8% classification accuracy at the edge device was proposed in [24].

It should be noted that most of the research work focuses on detection accuracy but overlooks the trade-off between speed and hardware requirements. Although a portable flow cytometer that uses deep learning to detect Giardia lamblia cysts in water samples is reported, its data processing module still ran on a desktop PC [25]. Moreover, these are post-experimental processes and are challenging to apply in a single-board computer (SBC) platform in real time applications since data acquisition and storing large number of images for further morphology analysis are limited by memory capacity.

The closest work to ours is deemed to be the portable and computer vision-based detection platform reported in [26] which can automatically classify the results from saved data without the need of professionals. A Raspberry Pi-4 SBC has been used for computing and a control module is used for controlling the data processing. However, the software code complexity, detection per second and processing time are not reported nor discussed in the paper.

To sum up, the number of research papers directly comparable with our work are limited; the closest ones and their key features are summarized in Table 1 (in addition, comparative results are provided in Table 3, see Section III).

**TABLE 1. State-of-the-art for blur free image acquisition and portable IFC vs. our work. complexity, portability, and cost are rated from low/no to high/yes (I.E. + to + + +).**

| Techniques | Complexity | Portability | Throughput/ accuracy | Cost | Summary of the methodologies |
|---|---|---|---|---|---|
| Virtual-freezing [21] | +++ | + | Throughput >10,000 cells/s; classification accuracy 95.3% | +++ | Acquires blur free single cell image by using a sophisticated system that includes a flow-controlled microfluidic chip, a speed-controlled polygon scanner, and a series of precise timing control circuits. |
| Deep learning [22] | + | + | 96.6% classification accuracy | ++ | Data processing runs on the computer |
| Machine learning [24] | +++ | ++ | Detection rate 93.8%; classification rate 99.8% | +++ | Optofluidic detection at the edge device relies in the time trace |
| Deep learning [25] | + | +++ | Autofocus on a single object in ~7 ms | ++ | Data processing runs on the computer |
| OpenCV [26] | + | +++ | Classification and counting accuracy are above 95% | + | The detection and counting system operate after saving 10 videos (RPI4 platform). However, the processing time for each object detection is not reported in the paper. |
| Proposed method | + | +++ | Counting accuracy is 100% for the tested videos | + | Acquires blur free single droplet image by using a lightweight pipeline on embedded platform |

To summarize, existing IFC systems are for laboratory grade applications and exhibit high complexity and high cost in order to perform droplet detection. Moreover, the captured blur-free images are transferred to desktop PCs for analysis, which is not suitable for portable devices. The development of a lightweight and reliable pipeline remains a crucial open issue for reducing the software code complexity so that it can run on an embedded platform. To bridge this gap, we built

a cost-effective pipeline which can acquire blur-free images and count each droplet automatically on a portable platform, without a laboratory environment.

### A. CONTRIBUTION

The main goal of this work is to acquire blur-free single droplet images without any camera synchronization to handle the complexity and computational cost challenge [27], [28].

Existing systems rely on complex hardware setups comprising high frame rate cameras that are not cost effective and complex optical solutions for droplet focusing. In contrast, we present a pipeline for acquiring single-droplet images that utilizes minimum hardware and a lightweight algorithm for detecting and acquiring single object images from video stream. In addition to single-image acquisition, our pipeline can count the total number of generated droplets. The proposed pipeline adopts color-based detection and a Euclidian distance tracker for droplet tracking in real time.

Consequently, our pipeline can be deployed on different platforms ranging from high-performance device (desktop PC) to hardware-accelerated device (Nvidia Jetson Nano) and to resource-constrained device (RPI4) for acquiring blur free single object image and counting the total number of objects. Experiments using video of fast-moving droplets (the input fluid flow rate was as high as 67.7 $\mu$L/min) show that the proposed pipeline achieves 2 ms, 25 ms and 10 ms processing time for each droplet on Desktop PC, Raspberry Pi-4 and Nvidia Jetson Nano, respectively. This yields a maximum of 500 DPS on a desktop PC and 100 DPS on a resource-constrained platform, which is fast enough to be utilized for high volume sample analysis where droplet generation rate lies to 100 to 500 Hz in IFC [29]. As counting and acquisition of single droplet image are both accomplished using lightweight detection and tracking algorithm, it is insensitive to camera synchronization. As multiple droplets can be detected simultaneously, the overall setup does not require an additional subsystem such as velocity detector to capture images of droplets under high flow rate. We also conducted experiments using external reference data (see Section III-C) to validate the proposed pipeline.

This proposed pipeline can automatically acquire blur-free single droplet image and subsequent droplet counting on resource constrained embedded platform at a rate of 100 DPS. Therefore, we believe that the pipeline presents a promising solution for future POC diagnostics, more specifically in object classification in droplets, e.g. for antimicrobial susceptibility testing, where the significantly higher frequency droplet-based assays could significantly contribute to sepsis prevention. At present, droplet-based imaging flow cytometry platforms primarily rely on high-end, high-power instrumentation (e.g. GPUs), which limits embedded or portable applications and scalability due to high associated component costs.

The rest of this paper is organized as follows: Section II presents the experimental setup including the materials used, the proposed detection and tracking method and its complexity analysis, and the process to acquire blur-free single droplet image acquisition along with counting. Section III presents the experimental results, and their analysis and Section IV provides a comparison against the closest related works and discussion thereof. Finally, Section V summarizes the key findings and limitations of this work and outlines possible future work.

## II. MATERIALS AND METHODS

### A. DROPLET GENERATION VIDEOS

The proposed pipeline has been tested firstly with simulated two-phase flow videos and secondly with real-world experimental fluorescent videos. Before explaining how these simulated and real-word videos are obtained, we first briefly introduce the physical system; the microfluidic droplets are generated inside a polydimethylsiloxane (PDMS) chip (schematic shown in Figure 1(a)). The microfluidic chip design is based on the real-life microfluidic chip [30] used in previous experiments. The chip contains water and oil inlets, a gas spring connector (denoted air inlet in Figure 1(a)), a flow-focusing junction with a width of 90 $\mu$m and a height of 100 $\mu$m, and an outlet. The oil inlets are equipped with filters that capture any particles that may be present in mineral oil (Sigma Aldrich 330779) in order to prevent the microchannels from becoming clogged.

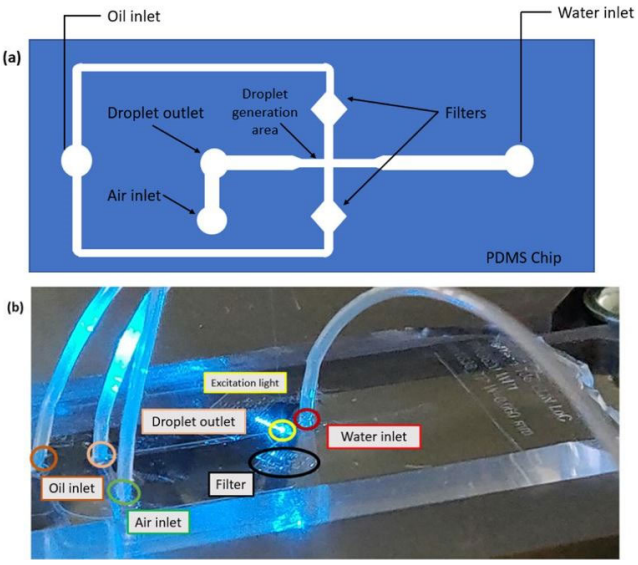


**FIGURE 1. Droplet generation. (a) Schematic of the PDMS chip (not drawn to scale). (b) Experimental setup showing all (water, oil, air) inlets, the droplet outlet, the filter, and the excitation laser light. The power supply for driving the laser, the computer for controlling the syringe pump, and the Basler Ace camera are not visible here.**

Firstly, the simulated videos are obtained from a two-phase flow simulation model (implemented in COMSOL Multiphysics version 5.6 in our previous work [30]), where water and oil flows are maintained at 10 $\mu$L/min and 67.7 $\mu$L/min, respectively, to generate droplets. The flow rate of oil is 6.77 times greater than that of water; this helps to keep

the droplets separated (relatively large distances between droplets make it easier for the simulation to avoid ''jetting'' in the channel). The liquid phases in COMSOL (see Fig. 3(A)) are displayed as red and blue. The central color, green, indicates the region where the relative concentration of water is 50%.

Secondly, for the experimental video, deionized water mixed with 10 $\mu$g/ml fluorescein isothiocyanate- dextran (FITC) and oil through corresponding inlets are used in the fabricated PDMS chip channel. The mixture and oil are injected through the chip inlet via a 3 mL syringe. The flow rates of the water and oil fluid are maintained by using a syringe pump (SpinSplit Netpump, SpinSplit LLC, Budapest) and air inlet has not been used in our experiment. As shown in Figure 1(b), a 488 nm laser light (Sharp GH04850B2G with focusing optics from AliExpress) is used as the excitation light source; it is located right after the junction where the droplets from [31].

In the experiments, the droplet flow through a PDMS chip is captured by a Basler Ace camera at 1000 frames per second (fps).

### B. DROPLET DETECTION AND TRACKING

To implement the imaging pipeline, custom code utilizing color-based detection and a Euclidean distance tracker algorithm are combined. The pipeline was developed in Python 3.8 using OpenCV 4.5.5.

The acquired imaging results were exported automatically in suitable formats (e.g.,.tiff,. png,.gif,.jpeg, and.bmp), and they could be used for further data analysis involving, for example, classification, or morphology analysis. The pipeline is also able to simultaneously count the total number of droplets during the time of detection.

Figure 2 shows the three main steps of the pipeline: (1) video streaming; (2) processing; and (3) result acquisition. The pipeline begins by acquiring the video input in real time and then processes it to obtain single-droplet images. This processing step can be divided into two main parts: detection and tracking. During tracking, the pipeline can also count the total number of detected droplets. Finally, the results are saved. What follows presents the details of the detection and tracking methods implemented in our pipeline.

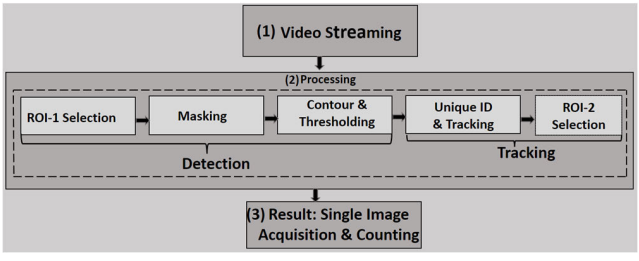


**FIGURE 2.** Block diagram of the single-image acquisition process flow, consisting of (1) video streaming, which occurs at 1000 fps; (2) processing, which involves detecting droplets based on their color and tracks; and (3) result acquisition, which involves acquiring a single image and counting the number of droplets using the tracking result without any duplication.

We conducted experiments with both the color-based detection and Background Subtraction Mixture of Gaussians-2 (BS-MOG2) methods. The color-based detection method is more efficient than BS-MOG2 algorithm because it does not fail if there is intensity variation in the experiments [32], [33]. Indeed, BS-MOG2 requires history and threshold input values, which in our case were selected to be 100 and 50, respectively, for the best optimization of the results. This history value defines how many last frames would affect the background model. BS-MOG2 creates a binary mask, where 0 represents the background and 1 represents the foreground, which works well for simulated video but fails in an experimental environment [34]. On the other hand, in our experiments, color-based detection works well for both environments. Mask was created by identifying the high and low range of the object color using HSV (Hue Saturation Value) range. To exclude the noise in both algorithms, the contours of the moving objects on the mask were passed through a threshold value (100 pixels), see Figure 5. Any objects above the threshold value were delimited by a bounding box. Once the object was detected inside the bounding box, each center position of the bounding rectangle was inserted in a single array.

Next, all the positions were fed to the tracker to update the unique ID for each droplet; a Euclidean distance tracker algorithm has been used to track each droplet. When a new droplet enters into region of interest-1 (ROI-1), the pipeline checks the central distance between consecutive droplets. An object generally follows a trajectory made up of points that are very close to each other. The closer the previous point is to the object, the greater the probability that the previous position belongs to it; in this case the tracker considers a droplet having a small distance ($< 10$ pixels) as the same object instead of new object. Therefore, it does not change the unique ID for this specific droplet. The algorithm calculates the Euclidean distance as follows [35]:

$$D = min(\sqrt{(a_j - a_i)^2 + (b_j - b_i)^2}) \quad (1)$$

where $D$ is the Euclidean distance of the previous tracks and the current tracks, $a$ is the x- coordinate of the track box, $b$ is the y-coordinate of the track box, $j$ represents all the prediction boxes in current frame, and $i$ represents all the prediction boxes in the previous frames.

The main purpose of tracking is to ensure precise droplet counting by preventing duplicate counting of droplets and to perform single-image acquisition at the same time. The tracking yields every single-droplet image without any duplication and counts the total number of droplets. The process of tracking a droplet consists of assigning it a unique ID that it retains until it leaves ROI-1. The algorithm performed well near the droplet generation area, where the single flow of droplets does not become occluded. However, after a short while, when the droplets reach the end of the channel,

they become occluded and lose their unique IDs. To solve this issue, and to save images as well as count the number of droplets, another region, ROI-2 (shown in Figure. 3), has been selected.

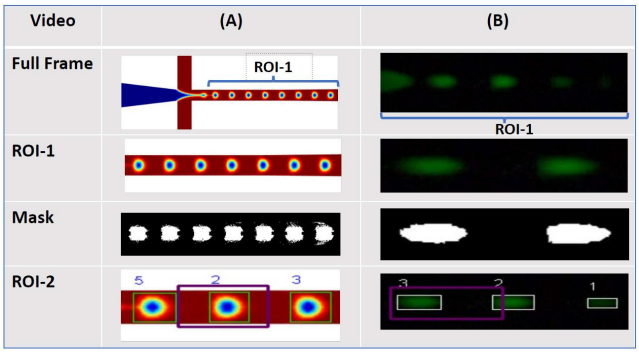


**FIGURE 3. Detailed video processing diagram showing the full frame, ROI-1, mask, and ROI-2. (A) COMSOL-simulated 2-phase flow video and (B) experimental fluorescence video captured by the Basler Ace camera at 1000 fps. For both cases, full frame shows the droplet generation inside the chip channel; ROI-1 represents only the specific small portion of the channel where the droplet flow can be observed; mask shows the object in white and the background in black (the droplet is flowing from left to right); and ROI-2 represents the small initial portion of the channel from the left side (purple region) used to obtain true counting.**

### C. IMAGING PIPELINE'S PSEUDO CODE AND COMPLEXITY ANALYSIS

This section illustrates the imaging pipeline's pseudo code and its time and space complexity analysis.

### D. TIME AND SPACE COMPLEXITY ANALYSIS

The time complexity of the pipeline depends on the dominant factors (video processing loop, masking, contour detection, contour filtering, and object tracking) contributing to the overall running time.

The primary loop iterates over each frame (denoted by $n$) of the video. The time complexity associated with each pixel in the frame can be approximated, with the big $O$ notation, as $O\ (m \times p)$ where $m$and $p$, *are* the frame dimensions. The time complexity for contour filtering depends on the number of contours and expressed as $O\ (c \times f)$ where $c$ is the number of contours and $f$ represents the time complexity associated with the operation within the contouring filter. $O\ (t \times g)$ is the time complexity for tracking where $t$ is the number of tracked object and $g$ represents the time complexity associated with the operations within the object tracking loop. Hence, the overall time complexity can be expresses as $O\ (n\times\ (m \times p)+ c \times f + t\times\ g)$.

The space complexity for each frame (input data) is constant as it is processed one at a time, and the total space complexity for handling video frames is negligible. The Euclidean distance tracker maintains a dictionary to store the center positions of tracked droplets and an integer to keep track of droplet IDs. If $s$ is the number of tracked droplets, then the space complexity of the tracker is $O(s)$. As the counting area polygon has a constant number of vertices the space complexity is $O\ (1)$. Moreover, if the total number of unique droplets detected is denoted by $q$, the space complexity is $O(q)$. A dictionary is needed to store images of detected droplets, so the space complexity depends on the number of detected objects $O(q)$. Various variables used for masking, contour detections, and bounding boxes have space

**Pseudo Code for Proposed Pipeline**

Data: Video frame
Result: Acquire single droplet image and count
1. tracker = EuclideanDistTracker()// Initialize Euclidean distance tracker
2. create_directory('data') // Create a directory named 'data' for saving droplet images
3. object_count = set() // Initialize droplet count set
4. Process frames in a loop
while True:
frame = read_frame(cap) // Read a frame from the video
Hsv_img = convert_to_hsv(frame) // Convert the frame to HSV
mask_green = create_green_mask(Hsv_img, Low_Green, High_Green) // Create mask using color thresholding
contours = find_contours(mask_green) // Find contours in the mask
detections = detect_objects(contours) // Detect droplets based on contour area
boxes_ids = tracker.update(detections, 1) // Update droplet tracking using the Euclidean distance tracker
5. Process each detected and tracked object
for box_id in boxes_ids:
x, y, w, h, obj_id = box_id
draw_rectangle(frame, x, y, w, h) // Draw bounding box
draw_text(frame, str(obj_id), x, y) // Text on the frame
result = point_in_polygon(roi_polygon, (x, y)) // Check if the droplet is inside the defined counting area
if result:
object_count.add(obj_id) // Update droplet_count based on the object ID
save_droplet_image(frame, obj_id) // Save image of the detected droplet
total_count, objects_in_second = calculate_counts(object_count, fps_count) // Calculate total droplet count and droplets per second
display_counts(frame, total_count) // Display the droplet count on the frame
7. Check for user input to exit the loop
key = wait_for_key()
if key == 27:
break
8. Cleanup and release the video capture object
release_video_capture(cap) // Release the video capture object
destroy_all_windows() // Destroy all open windows
End of Algorithm

complexity which is proportional to the size of the data, and these are temporary, hence do not contribute significantly to overall space complexity. Therefore, the overall space complexity is dominated by the Euclidean distance tracker i.e. $O(s)$ and the sets used for counting objects i.e. $O(q)$. Finally, the overall space complexity can be expressed as $O(s+q+1)$.

Memory usage in Figure 4 shows that it does not exhibit any significant spikes or drops throughout the overall processing. It takes approximately 95 Mebibytes (MiB) maximum.

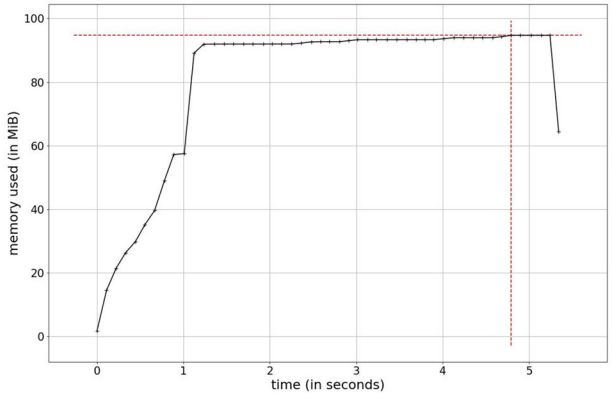


**FIGURE 4.** Memory consumption as a function of time. The horizontal axis shows the time of overall processing for whole video and the vertical axis shows memory consumption trend in Mebibytes.

The time complexity of the closest related paper [25] depends on the dominant factors (Background difference method, Gaussian filtering, Binarization, Opening and Expansion processing). For their case, let's assume $b$ is the number of pixels, so time complexity would be $O(b)$ for background difference. If $v$ is the size of gaussian kernel, time complexity would be for gaussian filtering would be $O(b \times v^2)$. The time complexity for binarization depends on the number of pixels in each image, hence $O(b)$. If $h$ is the number of pixels in the structural element time complexity for opening and expansion process would be $2 \times O(b \times h)$. The overall time complexity would be $O(b) + O(b \times v^2) + O(b) + 2 \times O(b \times h)$ which simplifies to $O(b \times (1 + v^2 + 2 \times h))$.

The memory required for storing the difference image, filtered image, binarized image, opening operation and expansion processing is proportional to the size of each frame, resulting $O(b)$ where $b$ is the total number of pixels in each frame. Table 2 shows the comparison of computational complexity of the proposed work and the closest related paper. The complexities are not provided in earlier works and hence are derived based on materials available in their paper.

In our proposed method, the dominant terms are $n$, $m$and $p$, while the second and third terms are constants and multiplied by $f$and $g$, respectively. On the other hand, the time complexity of the method in [26] involves the product of four variables which makes it more computationally complex, while its space complexity is lower.

**TABLE 2.** Comparison of computational complexity with closest related paper.

| Reference | Time Complexity | Space Complexity |
|---|---|---|
| OpenCV [25] | $O(b \times (1 + v^2 + 2 \times h))$ | $O(b)$ |
| Proposed Method | $O(n \times (m \times p) + c \times f + t \times g)$ | $O(s+q+1)$ |

### E. SINGLE-DROPLET IMAGE ACQUISITION AND COUNTING

In this process, a single-droplet image is acquired by automatically cropping each detected droplet from the bounding box. The location of one corner (instead of all four corners) of the bounding box is stored in a dictionary to reduce the complexity of the computation, and the object image is cropped when the particle is exactly inside of a specific ROI-2 (shown in Figure 5).

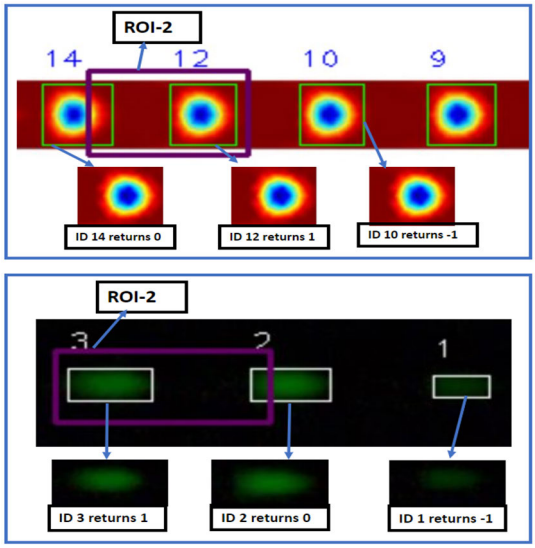


**FIGURE 5.** Three different conditions for accurate counting (Top: COMSOL simulated two-phase flow video, bottom: experimental fluorescence video): (i) if the green bounding box is outside ROI-2, the operation returns -1; (ii) if the bounding box is exactly on the boundary of ROI-2, it returns 0; and (iii) if the bounding box is inside ROI-2, it returns +1, which is the right condition to store the location.

The geometric operation has been conducted for three different conditions: (i) if the bounding box is outside ROI-2, the operation returns -1; (ii) if the bounding box is exactly on the boundary of ROI-2, it returns 0; and (iii) if the bounding box is inside ROI-2, it returns +1.

The location is stored only if the operation returns +1; otherwise, the algorithm does not take any action. The algorithm checks whether a box with the same unique ID has already been stored or not. If the box has already been stored, the algorithm will not store it again, thus preventing duplication. Although the main contribution of this algorithm

is to acquire a blur-free single-droplet image from a fast flow, the counting of droplets is also performed in near real time. This entire process does not require any additional hardware to acquire single-droplet images or to perform counting.

### F. OVERVIEW OF THE COMPUTING PLATFORMS

We have implemented our pipeline on three different platforms, ranging from high-performance device (desktop PC) to hardware-accelerated device (Nvidia Jetson Nano) and to resource-constrained device (RPI4 Model B).

1. Desktop PC with an Intel i5-10210U processor (4 cores, 8 threads) running at 1.6 GHz and with 16 GB RAM. The OS is Windows 10.
2. Raspberry Pi 4, based on quad-core ARM Cortex-A72 processor clocked at 1.5GHz. It has 4 GB of RAM and is used in headless mode.
3. The Jetson Nano is also a compact device (NVIDIA Jetson NANO Developer Kit); it has a 128-core Maxwell GPU and a quad-core ARM A57 CPU1.434 GB of LPDDR4 [36].

We first tested the droplet image acquisition and droplet counting on the Windows-based PC and then on the two embedded platforms.

## III. RESULTS

To validate the proposed pipeline, we performed experiments using A) 2-phase simulated video, and B) experimental fluorescent droplet generation video (see Section II-A) The flow rate is 100 $\mu$l/min, and the camera frame rate is 1000 fps. The fluid flow is of high throughput, which can easily create motion blur when images are captured using a camera. Image quality degradation does not take place if the flow rate is low, but it can occur for a high throughput. The proposed pipeline can handle moving droplet videos to obtain images of good quality.

### A. DROPLET DETECTION AND TRACKING

Contour area selection from the mask resulted in successful droplet detection. White pixel values greater than 100 were considered above the threshold value for accurate droplet area detection. Values below the threshold were considered as noise and were not included in further processing.

### B. SINGLE-DROPLET IMAGE ACQUISITION AND COUNTING

Every detected droplet was tracked using its unique ID and counted. The total droplet count increases whenever a new object enters ROI-2. A 0.05 s video that contains 18 droplets was used for the experiment, and in each experiment 18 images were properly recorded. These numbers were validated through manual counting by three human observers. In addition, we observed that the processing time of our pipeline in two platforms for fast-moving droplet detection, counting and single-image capture.

The main goal of our proposed pipeline is capturing a single-droplet image from the near real-time fluid flow.

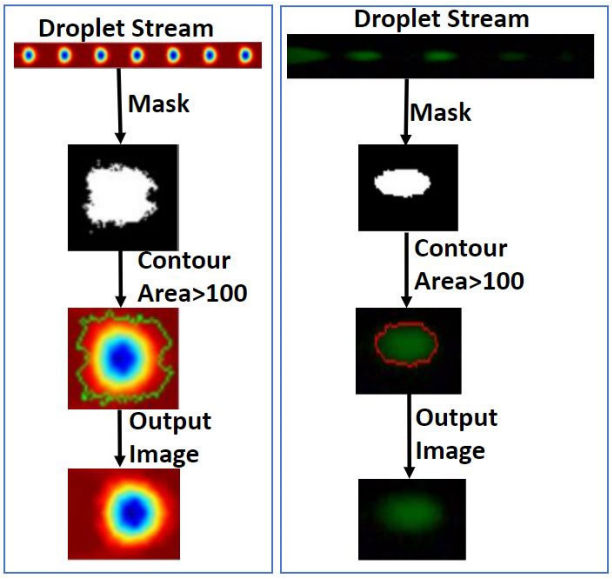


**FIGURE 6.** Manual thresholding if the contour area is greater than 100. Left: green contour for CFD (COMSOL) simulated video; right: red contour for experimental fluorescence video boundary. An area meeting the threshold criterion is considered to be an object, resulting in precise droplet detection.

The pipeline is suitable for capturing images of moving objects, not for capturing images of static objects. Since it can detect, count, and save each droplet successfully, it is suitable for any other similar single streaming IFC application.

The execution times for a single-droplet handling for the two different types of videos (simulated and experimental) using the two different algorithms (color-based and BS-MOG2) on the three platforms (desktop PC, RPI4, and Jetson Nano) are shown in Figure 7.

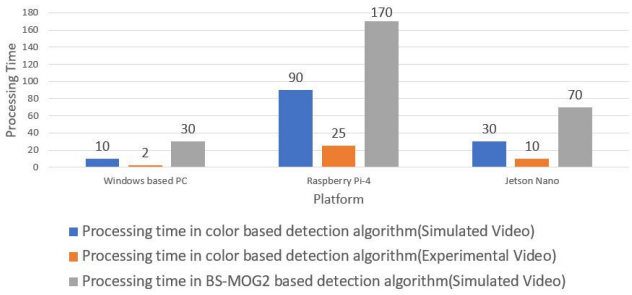


**FIGURE 7.** Comparison of single-frame processing time, where the horizontal axis represents the platforms, and the vertical axis represents the single-frame processing time in milliseconds(ms). The color corresponds to the algorithm and video type used. The color-based detection algorithm takes 10 ms, 90 ms, and 30 ms on the three different platforms, respectively, while the background subtraction algorithm takes 30 ms, 170 ms, and 70 ms for the simulated video. Experimental video yields the best result using color-based detection algorithm, which is 2 ms, 25 ms, and 10 ms on the desktop PC, Raspberry Pi 4, and Jetson Nano, respectively.

The color-based detection algorithm yields the minimum processing time for the experimental video on every platform; it takes 2 ms, 25 ms, and 10 ms on the desktop PC, Raspberry

Pi 4, and Jetson Nano, respectively. The same algorithm takes 3 to 5 times more time for the simulated video than it does for the experimental video (10 ms, 90 ms, and 30 ms, respectively).

Note that Figure 7 does not show results for BS-MOG2 on the experimental data; despite being adaptive and robust, the BS-MOG2 algorithm does not work on the experimental video. The reason is that BS-MOG2 is sensitive to Gaussian distribution of pixel and illumination variation. In this specific microfluidic droplet applications, the background is complex and dynamic because of the liquid flows and interactions. The single-droplet processing times with the BS-MOG2 algorithm on the simulated video are 30 ms, 170 ms, and 70 ms, for the Windows desktop PC, Raspberry Pi 4, and Jetson Nano, respectively.

Next, Table 3 shows the DPS and corresponding processing time for the two different algorithms on the two types of videos in three different platforms. As expected, due to its higher computational power, the desktop PC achieves the highest (500) DPS, with the color-based detection algorithm applied to the simulated video. However, it is also noticed that 100 DPS (10 ms to acquire one droplet image) can be obtained for the same algorithm and video with the Jetson Nano, which is considered as high throughput for the target applications.

**TABLE 3. Comparison of DPS on the three different platforms.**

| Algorithm | Data | Platform | Detected droplets per second (DPS) | Corresponding processing time (ms) |
|---|---|---|---|---|
| Color based detection | COMSOL Simulated Video | Windows-based desktop PC | 100 | 10 |
| | | Raspberry Pi-4 | 11 | 90 |
| | | Jetson Nano | 33 | 30 |
| | Experimental Video | Windows-based PC | 500 | 2 |
| | | Raspberry Pi-4 | 40 | 25 |
| | | Jetson Nano | 100 | 10 |
| BS-MOG2 based detection | COMSOL Simulated Video | Windows-based PC | 33 | 30 |
| | | Raspberry Pi-4 | 5 | 171 |
| | | Jetson Nano | 14 | 70 |

### C. TEST WITH EXTERNAL REFERENCE DATA

The pipeline was also tested with external reference brightfield microscopy video obtained with a similar chip design [37]. The microfluidic channel width was 50$\mu$m; the flow rates were controlled using neMESYS 290N syringe pumps (Cetoni) and gas-tight syringes (Hamilton) connected to PTFE tubing, and the generation of droplet was monitored using a high-speed Mini UX-100 camera (Photron). Single object image acquisition and counting works well using our proposed pipeline on this external reference data. Figure 8 illustrates the single droplet image acquisition in brightfield microscopy.

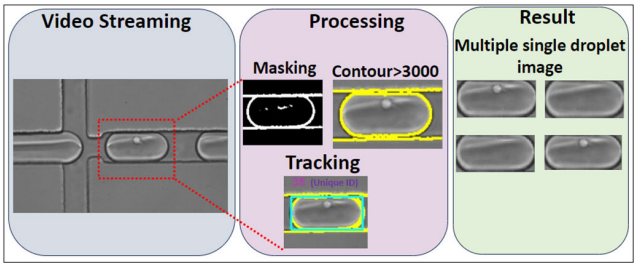


**FIGURE 8. Single droplet image acquisition in brightfield microscopy (Left to right: video streaming, processing, and result). Masking (white boundary), contouring (yellow boundary) and tracking (green rectangle) shows only for one droplet from the video. Multiple saved droplet images are shown in result.**

For automatic single droplet image acquisition and counting, the high and low range of the object color using HSV range needed to identify once to create mask. Then contour area selection from the mask was set with a threshold of 3000. This implies that white pixel values exceeding 3000 were taken into account the precise droplet area detection. Pixel values below this threshold were considered as noise and excluded from subsequent processing.

Each detected droplet was tracked using its unique ID and counted. The execution times for a single-droplet handling for brightfield microscopy image acquisition were not significantly different than that for the two-phase CFD simulation and fluorometry images as the overall process went through same steps.

## IV. COMPARISON AND DISCUSSION

To summarize, previous similar (i.e. closest related, but not directly comparable) studies for acquiring blur free images, detecting, and counting methodologies are either complex and not developed for being deployed into portable devices ([21], [37], [24]), or meant for other types of applications ([24]), or they do not provide all performance details [25]). In contrast, our proposed solution is able to acquire blur-free images, detect and count accurately each droplet automatically, making it suitable for portable platform.

These closest related works are summarized in Table 4 and further discussed.

Blur-free droplet image acquisition, detection, and counting the total number of droplets in an embedded platform has emerged as essential research towards POC technology. As indicated previously, the closest related works shown in Table 4 are not directly comparable, but for reference their platform suitability, processing time, and accuracy metrics were looked into. Detection and counting of microfluidic

**TABLE 4. Most-closely related results vs. our work.**

| References | Techniques | Platform | Processing time (ms) | Counting accuracy |
|---|---|---|---|---|
| [38] | Deep learning | High-end NVIDIA V-100 GPU | 33 | N/A |
| [21] | Sophisticated hardware setup | Desktop PC | NA | N/A |
| [25] | Deep learning | Nvidia RTX 2080 GPU | ~34 | N/A |
| [26] | OpenCV | Raspberry Pi | N/A | 95.96 to 99.7% |
| This work | Lightweight pipeline using color-based detection and Euclidean distance tracker | Jetson Nano | 10 | Counting accuracy is 100% for the tested videos |

droplets is obtained in [38] by utilizing YoloV5 which is fast enough to meet the requirement of imaging flow cytometry. However, this method has been optimized (processing time: 33 ms) for powerful GPU to meet the requirements of their specific application. A sophisticated hardware setup has been developed in [21] which can handle more than 10,000 cells/s throughput and acquire blur free images of cells (processing time no reported in their paper). However, the setup is for laboratory grade imaging flow cytometer. A deep learning based miniaturized imaging flow cytometer is developed for waterborne parasite detection which takes approximately 34 ms for autofocusing, color reconstruction and detection. Similar research was conducted targeting an embedded platform [26] by utilizing OpenCV for blood cell detection and classification but have not provided their processing time.

Contrary to existing systems that rely on additional hardware setup and complex techniques for blur-free image acquisition, object detection and counting which are often costly and not suitable for portable devices, our proposed pipeline is optimized for portability. It utilizes minimal hardware and lightweight color-based detection and Euclidian distance tracker for droplet tracking in real time algorithm for detecting, counting, and acquiring single object images from video stream. It requires approximately three times less processing time than the most closely related existing studies [25], [38], making it well suited for embedded platform. Memory complexity analysis of the proposed pipeline indicates a memory usage of 95 MiB, with most of it dedicated for storing the unique identification number for each droplet. The characteristics of the proposed work eliminate the need for complex and expensive hardware, thereby democratizing access to droplet microfluidic IFC.

## V. CONCLUSION

Single-droplet image acquisition is important for extracting valuable information about a given particle. Because of the high throughput and fast flow of IFC, researchers have focused on developing hardware systems to capture blur-free single images. The pipeline proposed in this study enables near real-time image acquisition without the integration of any additional hardware components; video streaming is performed at 1000 fps in this paper.

The pipeline, which consists of a color-based detection algorithm, is capable of acquiring high quality single droplet image and counting the total number of droplets from a near real-time, single-line high throughput droplet flow. The pipeline's output can be fed into any other algorithm for further analysis; it can be used to perform tasks such as classification, segmentation, or morphology analysis. Droplet detection is challenging when the droplets have a high velocity and occlusions happen after some period, but the proposed pipeline is reliable enough to solve these issues.

The complete process is automated, and it can detect and count droplets, as well as acquire blur free single-droplet image in near real time. Identification of low and high color ranges of the object to create mask as well as setting manual threshold is used only once in initial phase. Despite the constraint of limited data, the proposed pipeline was tested using external data for validation. The pipeline was tested on a desktop PC and two embedded platforms, and the processing times of two different object detection algorithms were compared. The achievable droplet detection per second value is 100 DPS on the Jetson Nano SBC. The processing time on desktop PC is approximately five times faster; hence, it yields 500 DPS. For future work, an extension to this work would be to adapt auto-thresholding and the deployment of the pipeline in other embedded platforms to assess and compare their performances. Moreover, real time-analysis, exploration of hardware and software optimizations for possibly improving the performance and/or efficiency of the implementation is a future research direction and extension of the proposed work.

## ACKNOWLEDGMENT

The authors would like to thank to Assist. Prof. Tomasz Kaminski, University of Warsaw, Poland, for providing us with the validation data.

## REFERENCES

[1] D. A. Basiji, W. E. Ortyn, L. Liang, V. Venkatachalam, and P. Morrissey, "Cellular image analysis and imaging by flow cytometry," *Clinics Lab. Med.*, vol. 27, no. 3, pp. 653–670, Sep. 2007, doi: 10.1016/j.cll.2007.05.008.

[2] Y. Han, Y. Gu, A. C. Zhang, and Y.-H. Lo, "Review: Imaging technologies for flow cytometry," *Lab Chip*, vol. 16, no. 24, pp. 4639–4647, 2016, doi: 10.1039/c6lc01063f.

[3] B. S. Edwards and L. A. Sklar, "Flow cytometry: Impact on early drug discovery," *SLAS Discovery*, vol. 20, no. 6, pp. 689–707, Jul. 2015, doi: 10.1177/1087057115578273.

[4] M. Ding and B. S. Edwards, "High-throughput flow cytometry in drug discovery," *SLAS Discovery*, vol. 23, no. 7, pp. 599–602, Aug. 2018, doi: 10.1177/2472555218778250.

[5] T. Nguyen, V. A. Chidambara, S. Z. Andreasen, M. Golabi, V. N. Huynh, Q. T. Linh, D. D. Bang, and A. Wolff, "Point-of-care devices for pathogen detections: The three most important factors to realise towards commercialization," *TrAC Trends Anal. Chem.*, vol. 131, Oct. 2020, Art. no. 116004.
[6] A. K. Pomerantz, F. Sari-Sarraf, K. J. Grove, L. Pedro, P. J. Rudewicz, J. W. Fathman, and T. Krucker, "Enabling drug discovery and development through single-cell imaging," *Exp. Opinion Drug Discovery*, vol. 14, no. 2, pp. 115–125, Feb. 2019, doi: 10.1080/17460441.2019.1559147.
[7] J. Xu, D. Zhou, D. Deng, J. Li, C. Chen, X. Liao, G. Chen, and P. A. Heng, "Deep learning in cell image analysis," *Intell. Comput.*, vol. 2022, pp. 1–15, Sep. 2022, doi: 10.34133/2022/9861263.
[8] M. Doan, I. Vorobjev, P. Rees, A. Filby, O. Wolkenhauer, A. E. Goldfeld, J. Lieberman, N. Barteneva, A. E. Carpenter, and H. Hennig, "Diagnostic potential of imaging flow cytometry," *Trends Biotechnol.*, vol. 36, no. 7, pp. 649–652, Jul. 2018, doi: 10.1016/j.tibtech.2017.12.008.
[9] Z. Liu, W. Guo, D. Zhang, Y. Pang, J. Shi, S. Wan, K. Cheng, J. Wang, and S. Cheng, "Circulating tumor cell detection in hepatocellular carcinoma based on karyoplasmic ratios using imaging flow cytometry," *Sci. Rep.*, vol. 6, no. 1, p. 39808, Dec. 2016, doi: 10.1038/srep39808.
[10] A. Lopresti, F. Malergue, F. Bertucci, M. L. Liberatoscioli, S. Garnier, Q. DaCosta, P. Finetti, M. Gilabert, J. L. Raoul, D. Birnbaum, C. Acquaviva, and E. Mamessier, "Sensitive and easy screening for circulating tumor cells by flow cytometry," *JCI Insight*, vol. 4, no. 14, Jul. 2019, Art. no. e128180, doi: 10.1172/jci.insight.128180.
[11] E. Shechtman, Y. Caspi, and M. Irani, "Increasing space-time resolution in video," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 2350, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., Berlin, Germany: Springer, 2002, pp. 753–768, doi: 10.1007/3-540-47969-4_50.
[12] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich, "Synchronization and rolling shutter compensation for consumer video camera arrays," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Miami, FL, USA, Jun. 2009, pp. 1–8, doi: 10.1109/CVPRW.2009.5204340.
[13] D. V. Voronin, A. A. Kozlova, R. A. Verkhovskii, A. V. Ermakov, M. A. Makarkin, O. A. Inozemtseva, and D. N. Bratashov, "Detection of rare objects by flow cytometry: Imaging, cell sorting, and deep learning approaches," *Int. J. Mol. Sci.*, vol. 21, no. 7, p. 2323, Mar. 2020, doi: 10.3390/ijms21072323.
[14] W. E. Ortyn, B. E. Hall, T. C. George, K. Frost, D. A. Basiji, D. J. Perry, C. A. Zimmerman, D. Coder, and P. J. Morrissey, "Sensitivity measurement and compensation in spectral imaging," *Cytometry A*, vol. 69, no. 8, pp. 852–862, Aug. 2006, doi: 10.1002/cyto.a.20306.
[15] E. K. Zuba-Surma, M. Kucia, A. Abdel-Latif, J. W. Lillard, and M. Z. Ratajczak, "The ImageStream System: A key step to a new era in imaging," *Folia Histochemica et Cytobiologica*, vol. 45, no. 4, p. 13, 2007.
[16] A. S. Rane, J. Rutkauskaite, A. de Mello, and S. Stavrakis, "High-throughput multi-parametric imaging flow cytometry," *Chem*, vol. 3, no. 4, pp. 588–602, Oct. 2017, doi: 10.1016/j.chempr.2017.08.005.
[17] E. Schonbrun, S. S. Gorthi, and D. Schaak, "Microfabricated multiple field of view imaging flow cytometry," *Lab Chip*, vol. 12, no. 2, pp. 268–273, 2012, doi: 10.1039/c1lc20843h.
[18] A. K. S. Lau, A. H. L. Tang, J. Xu, X. Wei, K. K. Y. Wong, and K. K. M. Tsia, "Optical time stretch for high-speed and high-throughput imaging—From single-cell to tissue-wide scales," *IEEE J. Sel. Topics Quantum Electron.*, vol. 22, no. 4, pp. 89–103, Jul. 2016, doi: 10.1109/JSTQE.2015.2512978.
[19] B. Guo, C. Lei, H. Kobayashi, T. Ito, Y. Yalikun, Y. Jiang, Y. Tanaka, Y. Ozeki, and K. Goda, "High-throughput, label-free, single-cell, microalgal lipid screening by machine-learning-equipped optofluidic time-stretch quantitative phase microscopy," *Cytometry A*, vol. 91, no. 5, pp. 494–502, May 2017, doi: 10.1002/cyto.a.23084.
[20] K. Goda, A. Ayazi, D. R. Gossett, J. Sadasivam, C. K. Lonappan, E. Sollier, A. M. Fard, S. C. Hur, J. Adam, C. Murray, C. Wang, N. Brackbill, D. Di Carlo, and B. Jalali, "High-throughput single-microparticle imaging flow analyzer," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 29, pp. 11630–11635, Jul. 2012, doi: 10.1073/pnas.1204718109.
[21] H. Mikami, M. Kawaguchi, C. J. Huang, H. Matsumura, T. Sugimura, K. Huang, C. Lei, S. Ueno, T. Miura, T. Ito, and K. Nagasawa, "Virtual-freezing fluorescence imaging flow cytometry," *Nature Commun.*, vol. 11, no. 1, p. 1162, Mar. 2020, doi: 10.1038/s41467-020-14929-2.
[22] Y. Wang, Z. Huang, X. Wang, F. Yang, X. Yao, T. Pan, B. Li, and J. Chu, "Real-time fluorescence imaging flow cytometry enabled by motion deblurring and deep learning algorithms," *Lab Chip*, vol. 23, no. 16, pp. 3615–3627, 2023.
[23] R. K. R. Baskaran, A. Link, B. Porr, and T. Franke, "Classification of chemically modified red blood cells in microflow using machine learning video analysis," *Soft Matter*, vol. 20, no. 5, pp. 952–958, 2024.
[24] V. Ganjalizadeh, G. G. Meena, M. A. Stott, A. R. Hawkins, and H. Schmidt, "Machine learning at the edge for AI-enabled multiplexed pathogen detection," *Sci. Rep.*, vol. 13, no. 1, p. 4744, Mar. 2023.
[25] Z. Göröcs, D. Baum, F. Song, K. de Haan, H. Ceylan Koydemir, Y. Qiu, Z. Cai, T. Skandakumar, S. Peterman, M. Tamamitsu, and A. Ozcan, "Label-free detection of giardia lamblia cysts using a deep learning-enabled portable imaging flow cytometer," *Lab Chip*, vol. 20, no. 23, pp. 4404–4412, Nov. 2020.
[26] R. Wang, X. Huang, X. Xu, J. Sun, S. Zheng, X. Ke, J. Yao, W. Han, M. Wei, J. Chen, H. Gao, J. Guo, and L. Sun, "A standalone and portable microfluidic imaging detection system with embedded computing for point-of-care diagnostics," *IEEE Sensors J.*, vol. 22, no. 6, pp. 6116–6123, Mar. 2022.
[27] S.-Y. Chiu, C.-C. Chiu, and S. S.-D. Xu, "A background subtraction algorithm in complex environments based on category entropy analysis," *Appl. Sci.*, vol. 8, no. 6, p. 885, May 2018, doi: 10.3390/app8060885.
[28] N. G. El-Gamal, H. E. Moustafa, and F. E. Z. Abou-Chadi, "A new combination method for background subtraction in video sequences," in *Proc. 8th Int. Conf. Informat. Syst. (INFOS)*, May 2012, pp. 1–25.
[29] A. Lashkaripour, C. Rodriguez, L. Ortiz, and D. Densmore, "Performance tuning of microfluidic flow-focusing droplet generators," *Lab Chip*, vol. 19, no. 6, pp. 1041–1053, 2019.
[30] K. Pärnamets, A. Udal, A. Koel, T. Pardy, N. Gyimah, and T. Rang, "Compact empirical model for droplet generation in a lab-on-chip cytometry system," *IEEE Access*, vol. 10, pp. 127708–127717, 2022, doi: 10.1109/ACCESS.2022.3226623.
[31] *Diy 12×40 mm 5.6 mm To18 Laser Diode Housing Case W/405 nm 450 nm 650 nm 780 nm 980 nm Collimating Lens*. Accessed: Jan. 15, 2023. [Online]. Available: http://www.aliexpress.com/item/1005003509604067.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=
[32] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit.*, Cambridge, U.K., 2004, pp. 28–31, doi: 10.1109/ICPR.2004.1333992.
[33] I. Iszaidy, R. Ngadiran, R. B. Ahmad, N. Ramli, M. I. Jais, and V. Vijayasarveswari, "An analysis of background subtraction on embedded platform based on synthetic dataset," *J. Phys., Conf.*, vol. 1755, no. 1, Feb. 2021, Art. no. 012042, doi: 10.1088/1742-6596/1755/1/012042.
[34] Y. Xu, H. Ji, and W. Zhang, "Coarse-to-fine sample-based background subtraction for moving object detection," *Optik*, vol. 207, Apr. 2020, Art. no. 164195, doi: 10.1016/j.ijleo.2020.164195.
[35] Y. Zhao, X. Zhou, X. Xu, Z. Jiang, F. Cheng, J. Tang, and Y. Shen, "A novel vehicle tracking ID switches algorithm for driving recording sensors," *Sensors*, vol. 20, no. 13, p. 3638, Jun. 2020, doi: 10.3390/s20133638.
[36] *NVIDIA Jetson NANO Developer Kit*.
[37] F. Hollfelder, "Microfluidics-enabled fluorescence-activated cell sorting of single pathogen-specific antibody secreting cells for the rapid discovery of monoclonal antibodies," to be published
[38] M. Durve, A. Tiribocchi, F. Bonaccorso, A. Montessori, M. Lauricella, M. Bogdan, J. Guzowski, and S. Succi, "DropTrack—Automatic droplet tracking with YOLOv5 and DeepSORT for microfluidic applications," *Phys. Fluids*, vol. 34, no. 8, Aug. 2022, Art. no. 082003.

**FARIHA AFRIN** (Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from International Islamic University Chittagong, Bangladesh, in 2013, the M.Sc. degree in electrical and electronics engineering from American International University-Bangladesh, in 2017, and the master's degree in communicative electronics from Tallinn University of Technology, Estonia, in 2020, where she is currently pursuing the Ph.D. degree with the Thomas Johann Seebeck Department of Electronics. Her research interests include fluorescence and morphology detection in flow cytometry.

**KAISER PÄRNAMETS** (Member, IEEE) was born in Tallinn, Estonia, in January 1989. He received the B.Sc. and M.Sc. degrees in engineering from Tallinn University of Technology, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree. From 2014 to 2017, he was a Hardware Engineer at the electronics industry. Since 2018, he has been an Electronics Lecturer for bachelor's and master's students. His research interests include lab-on-a-chip and microfluidics.

**YANNICK LE MOULLEC** (Senior Member, IEEE) received the M.Sc. degree from Université de Rennes I, France, in 1999, and the Ph.D. and HDR (accreditation to supervise research) degrees from Université de Bretagne Sud, France, in 2003 and 2016, respectively. From 2003 to 2013, he was a Postdoctoral Researcher, an Assistant Professor, and an Associate Professor with the Department of Electronic Systems, Aalborg University, Denmark. He joined the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, Estonia, first as a Senior Researcher, from 2013 to 2016, and a Professor, since 2017. He has supervised or co-supervised 60 M.Sc. theses and 15 Ph.D. theses. He has been involved in more than 20 projects, including five as a PI, a co-PI, or a co-main applicant. His research interests include embedded systems and wireless systems, including the IoT and the application thereof. He is a member of the IEEE Sustainable ICT Technical Community and the IEEE Circuits and Systems Society.

**ANDRES UDAL** (Member, IEEE) received the Ph.D. degree in electronical engineering from Tallinn University of Technology, Tallinn, Estonia, in 1999. From 1990 to 1995, he was acting as a Visiting Researcher and a TCAD Software Designer at Silvaco Int., Santa Clara, CA, USA; Darmstadt University of Technology, Germany; and Uppsala University, Sweden. He is currently a Senior Research Fellow with the Department of Software Science, Laboratory for Proactive Technologies, Tallinn University of Technology.

**ANTS KOEL** (Member, IEEE) was born in Tallinn, Estonia, in August 1962. He received the Engineering Diploma degree in industrial electronics from Tallinn Polytechnic Institute, Tallinn, Estonia, in 1985, and the master's and Ph.D. degrees from Tallinn University of Technology, in 1998 and 2014, respectively. His research interests include semiconductor electronics and devices, TCAD simulations, and circuit design. He was the Chairperson of the Steering Committee of the IEEE-Sponsored Baltic Electronics Conference, in 2020 and 2022, participated in ERA Chair projects COEL and 5GSOLAR, led electronics in-kind contribution projects of Estonia in ESS.

**TAMAS PARDY** (Member, IEEE) received the M.Sc. degree in info-bionics engineering from Pázmány Péter Catholic University, Budapest, Hungary, in 2014, and the Ph.D. degree in electronics and telecommunication from Tallinn University of Technology, Tallinn, Estonia, in 2018. He is currently a Senior Researcher with Tallinn University of Technology. He has supervised one Ph.D. thesis and three M.Sc. theses and has authored or co-authored 20 scientific publications. His research interests include flow- and temperature-control of lab-on-a-chip devices.

**TOOMAS RANG** (Senior Member, IEEE) received the Ph.D. degree in semiconductor electronics from the Hungarian Academy of Sciences, in 1981. He is currently a Professor Emeritus with the Thomas Johann Seebeck Department of Electronics and a Senior Research Fellow with the Department of Chemistry and Biotechnologies, Tallinn University of Technology, Estonia. He has supervised 15 Ph.D. students and has fulfilled the PI position in several European and domestic research and industrial projects. His current research interest includes applied microfluidics approaches for lab-on-chip applications. He is the Initiator and today the honor Chairperson of the Baltic Electronics Conference (BEC) Series, in 1987, supported by IEEE, since 1996.