





Improving the Classification Accuracy in Label-Free Flow Cytometry Using Event-Based Vision and Simple Logistic Regression

Muhammed Gouda , Alessio Lugnan , Joni Dambre , Gerd van den Branden , Christoph Posch , and Peter Bienstman , *Member, IEEE*

Abstract—Event-based cameras are novel bio-inspired vision sensors that do not follow the mechanism of traditional frame-based cameras. In the field of data acquisition, the replacement of CMOS cameras with event-based cameras has proved to enhance the accuracy of machine learning methods in situations where critical lighting conditions and rapid dynamics are paramount. In this paper, we investigate for the first time the use of extreme learning machines on data coming from event-based cameras in the context of flow cytometry. Except for the image sensor, the experimental setup is similar to a setup we used in (Lugnan et al., 2020) where we showed that a simple linear classifier can achieve around 10% error rate on background-subtracted cell frames. Here, we show that the error rate of this simple imaging flow cytometer could be decreased to less than 2% just by making use of the capabilities of an event camera. Moreover, additional benefits like more sensitivity and efficient memory usage are gained. Finally, we suggest further possible improvements to the experimental setup used to record events from flowing micro-particles allowing for more accurate and stable cell sorting.

Index Terms—Event-based camera, extreme learning machine, flow cytometry, image classification.

I. INTRODUCTION

FLOW cytometry is a technology used in the biomedical industry which targets separating different types of cells based on multiple characteristics [2], [3]. These characteristics may include size, shape, or fluorescence [4]. The technology has been applied in different areas, from the diagnosis of cancer and cancer treatment progression, to microparticles and microalgae classification [5], [6], [7], [8].

Manuscript received 25 May 2022; revised 8 February 2023; accepted 8 February 2023. Date of publication 10 February 2023; date of current version 7 March 2023. This work was supported in part by the European H2020 Neoteric Project under Grant Agreement 871330 and in part by Flemish Fund for Scientific Research (FWO-Vlaanderen) under Grant G006020N. (*Corresponding author: Muhammed Gouda.*)

Muhammed Gouda, Alessio Lugnan, and Peter Bienstman are with the Photonics Research Group, Ghent University - imec, 9052 Ghent, Belgium (e-mail: muhammedgoudaahmed.gouda@ugent.be; alessio.lugnan@ugent.be; peter.bienstman@ugent.be).

Joni Dambre is with the Photonics Research Group, IDLab, 9052 Ghent, Belgium (e-mail: joni.dambre@ugent.be).

Gerd van den Branden and Christoph Posch are with the Prophesee, 75012 Paris, France (e-mail: gvandenbranden@prophesee.ai; cposch@prophesee.ai).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSTQE.2023.3244040>.

Digital Object Identifier 10.1109/JSTQE.2023.3244040

Traditionally, the cells used in flow cytometry experiments are initially labeled with different biomarkers. However, labeling cells with biomarkers may lead to chemical interaction with the cell, changing its characteristics and ruining the validity of the results [9]. To avoid such problems, label-free (also called stain-free) imaging flow cytometry captures thousands of images by a CMOS camera from light diffracted by flowing cells which have not been labelled [10]. The images are then processed and sent to a traditional image processing or to a machine learning model which is trained to learn how to separate different classes of cells. In spite of their successes: we identify here two main issues with current imaging flow cytometers, the memory issue and the background noise issue.

First, the memory problem stems from the fact that a high-speed imaging flow cytometer (classifying thousands of cells per second) requires a camera with a high frame rate. This results in a huge amount of data. Second, the pixel values in the frames captured by a CMOS camera do not contribute directly as features for the machine learning model. This is because spatial information that is useful for the training model usually comes with additional noise from the background.

The noise issue has been tackled by simply filtering out the background signal. In Lugnan et al. [1], we subtracted each individual frame from the previous one and then multiplied individual pixels by training weights [1]. A more advanced noise filtering approach was followed in [11] where they designed a denoising autoencoder, which is a nonlinear unsupervised model, in order to increase the signal to noise ratio. Nonetheless, this filtering is never ideal and adds to the computational cost of the overall system.

The aim of this paper is to show that replacing the CMOS camera with an event-based camera will solve the two issues mentioned above, without relying on a noise or background filter. In order to do so, we will first need to introduce the concept of event vision.

The rest of this paper is structured as follows: in Section II, the theory behind event-based vision along with its recent applications in the literature are presented. The experiments performed to capture the particle images are illustrated in Section III. Then, in Section IV, the machine learning pipeline and the methodology followed to alleviate measurement bias, sometimes referred to as shortcut learning [12], are discussed.



Fig. 1. The event-based camera from Prophesee [15] used in this work is shown. The intensity of the light is sampled by the camera, and an event is fired whenever the intensity exceeds by a certain value. Based on whether there is an increase or a decrease in the intensity, a positive or a negative signal is recorded.

The accuracy of different models, a sanity check test and some examples of misclassifications are shown in Section V. Finally, Section VI discusses additional gains of the current system such as higher sensitivity and memory efficiency, and suggests further improvements.

II. EVENT-BASED VISION AND ITS SUCCESSES

Unlike traditional CMOS cameras, event-based cameras do not acquire consecutive static images at a certain frame rate [13]. The idea was introduced by [14] where a dynamic vision sensor consisted of pixels that respond independently to changes in the scene. Whenever the logarithm of the intensity of the light incident on pixel changes by a value exceeding a certain threshold as shown in Fig. 1. If this threshold is very small, a large number of unwanted events from the background will be recorded. On the other hand, having a threshold with a significantly high value will result in a loss of the targeted events. As an example, in our application, we would like this threshold to be high enough so that we do not capture speckle noise from the laser, and small enough to sense the variation of the diffraction pattern caused by the passing particles.

This way, an event camera only responds to the dynamics present in a scene. In other words, its working principle inherently filters out temporally-redundant information [9], [16], [17], [18]. Moreover, events can be generated on a time scale of microseconds making it suitable, for example, to record high-speed chemical kinetics [19]. The field of event-based vision is taking off, showing many recent successes in literature. For example, in [16], a regression task for predicting a steering wheel's angle has been studied. In their work, they showed that a convolutional neural network (CNN) taking features from an event-based camera outperformed the accuracy of similar CNN models with CMOS sensors, and that in multiple scenarios (day and evening). Examples from their datasets show that the event camera results in samples with significantly smaller motion blur compared to their CMOS counterparts. On top of this, sudden alterations in the lighting condition of a scene did not ruin the resulted frames. Other applications of event vision were presented in [20] and [21] where they recognized hand gestures

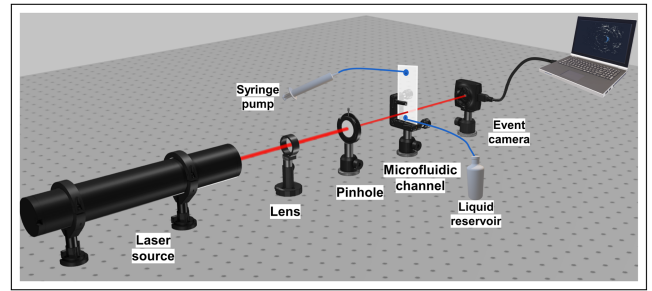


Fig. 2. The experimental setup built to generate the training and test datasets. Light coming from a 633 nm He-Ne laser is focused by a lens on a $25 \mu\text{m}$ pinhole. Behind the pinhole is a vertically-mounted PMMA microfluidic channel inside which microparticles are flowing downwards. The diffraction pattern caused by a flowing particle is captured by the event-based camera which is connected to a laptop with a dedicated software for recording the events fired at different pixels.

and human gait (walking style) respectively. More recently, there has been a number of showcases where event-based cameras were used to detect and track moving objects [22], [23], [24], [25], [26].

III. METHODS

The setup built in this work as shown in Fig. 2 consists of a laser source emitting light with 632.8 nm wavelength. After passing through a lens and a $25 \mu\text{m}$ pinhole, the light is focused on a PMMA microfluidic channel. Inside the channel, flowing microparticles are pumped using a manual syringe pump which is connected to the upper port, while a liquid reservoir is connected to the other port. The particles are pumped using 50 ml manual syringe pumps. We used three different syringes, one for particle A, another one for particle B and the third one for flushing water. A Prophesee event camera records the diffraction pattern caused by the flowing particles. We used two different classes of spherical micro-particles (class A of diameter $16 \mu\text{m}$ and class B with a diameter of $20 \mu\text{m}$). We purchased the particles from PolyAn GmbH company based in Germany.

We use 0.6 ml of particles and dilute the solutions with 30 ml water. Then, we add a drop of Triton X-100 surfactant to prevent the particles from clustering.

To generate the data-set for our model, we performed two separate experiments for each class of the particles. We start by flushing the inside of the microfluidic channel in both directions. After that, we pump particles from class A for 1 minute, then water is flushed again in both directions before class B particles are inserted.

IV. MEASUREMENT BIAS

A machine learning model can be easily biased by the measurement conditions. So, instead of learning the features that attribute directly to the size of the flowing particle, it might learn instead features that are associated with the condition of the experiment, and use these as a shortcut to achieve a good training score that does not generalise well during testing. In our particular case of particle interference pattern acquisition, because of the extremely high measurement sensitivity, it is difficult (if not

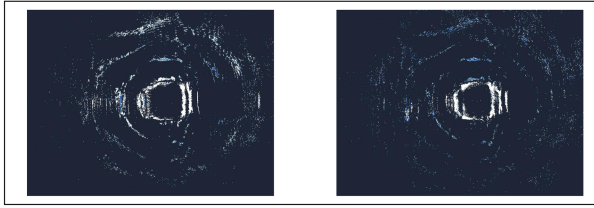


Fig. 3. Samples from the event-based camera using two classes of micro-particles. On the left is particle A with diameter $16 \mu\text{m}$, and on the right is particle B with diameter $20 \mu\text{m}$.

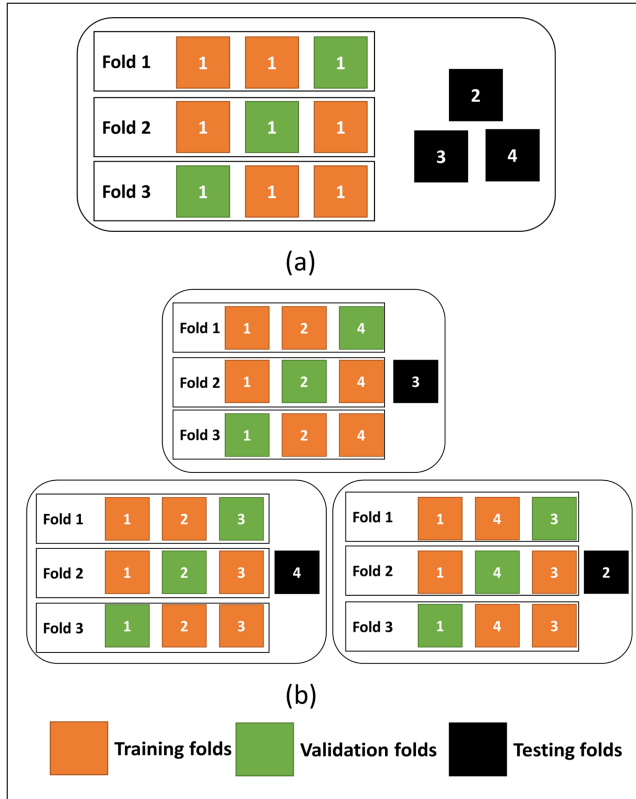


Fig. 4. Training-validation-test splits for (a) the single-session training model and (b) the intertwined models. The numbers in the boxes represent the measurement session number. In both cases, cross validation is implemented with 3 folds. In (a), the samples in the three folds are drawn from the first measurement session, while in (b) the validation fold is always drawn from a measurement session not seen in the two training folds.

impossible) to avoid data bias due to measurement conditions. In order to reduce the effect of measurement bias, we followed the intertwined measurement approach in Lugnan et al. [1] where the training data is drawn from multiple measurements during the day. In particular, we performed 4 iterations of the experiment explained above measuring the following particles: A1, B1, A2, B2, A3, B3, A4, B4. Fig. 4(a) shows the split methodology we first used. Here, we used samples from the first measurement session for training and validating the model (with 3-fold cross validation). Afterwards, the model is tested on data from other measurement sessions (measurements 2, 3 and 4). This single-session-training methodology, however, can be biased by measurement conditions of the first session,

therefore another approach to splitting the data was followed. In figure 4(b), we made sure that at each iteration, the samples in the validation folds are drawn from a measurement not used in the training folds. In addition, the test data are from a measurement not used in any of the training or validation folds. We will call this the intertwined model. In the next section, a comparison between the performance of these two data splitting approaches will be presented.

A. Machine Learning Pipeline

As mentioned in the previous section, the event-based camera does not generate frames of images that can be directly processed and sent to the machine learning algorithm. Rather, what we get from the camera is a sequence of events. Each event is a tuple with four elements (x, y, t, p) , where x and y are the spatial coordinates of the event, t is the time stamp and p is the polarity and it's either +1 or -1 depending on whether an increase of a decrease in intensity has occurred. We feed this sequence of events to a frame generation algorithm which takes as an argument the integration time during which the events are collected and plotted in a 2D frame. The integration time is a crucial parameter for our experiment. A value that is too low might result in frames that contains very little information (in other words, a frame might contain 3 events which happened at three different pixels in a relatively short time period and that does not account for the full presence of a particle). On the other hand, if we choose our accumulation time to be relatively large, we end up with frames saturated with many events happening at most of the camera's pixels, possibly corresponding to different particles.

Therefore, we estimate our integration time based on the speed of the flowing particles which can be calculated given the flow rate (3 ml/min), the cross-section area of the microfluidic channel ($200 \times 200 \mu\text{m}^2$) and the length of channel (58.5 mm). This results in a particle speed of 1.25 m/s. Knowing that the spot size of the laser is $25 \mu\text{m}$ in diameter, we calculate the travel time of the particle through the field of view to be:

$$t_{travel} = 20 \mu\text{s}$$

The concentration of the particles in the solution is

$$C = \frac{0.6 \text{ ml}}{30 \text{ ml}} = 2\%$$

Therefore, the time we use for the integration time is

$$t_{integration} = \frac{t_{travel}}{C} = \frac{20 \mu\text{s}}{2\%} = 10^3 \mu\text{s}$$

Another method we used to confirm this number was by looking at the evolution of the number of events as particles were passing. Before entering the field of view, the number of events generated is zero. Then this number starts to increase once the particle is captured entering the field of view, and afterwards it decreases again to zero when the particle exits. This generates a bell-curve which had a width of approximately 1 ms. Using this accumulation time, we could generate 80 K samples from all the experiments.

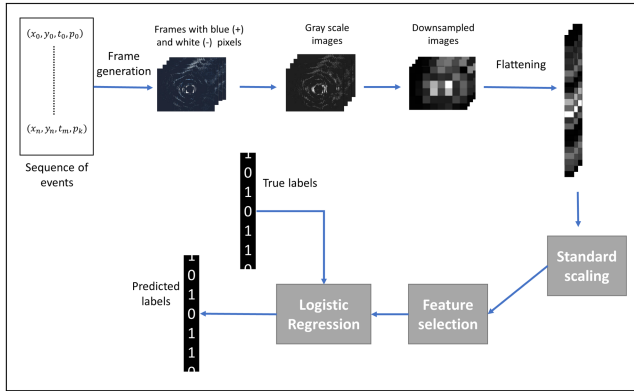


Fig. 5. Machine learning model's pipeline. Prior to applying a linear classifier, some preprocessing steps are performed. The events from the camera are processed by a frame generation algorithm which takes a certain accumulation time as input and outputs frames with positive and negative pixels. We then convert these frames into gray-scale images which are then flattened and stacked together to form a feature matrix. After that, a standardization and a feature selection steps are applied on the feature matrix before being used by the classifier.

The frames from the generation algorithm (which contains blue and white pixels based on the polarity of the corresponding event) were then converted into gray scale, therefore the polarity of the events became less expressed. Then, the images are downsampled. We used 6 different downsampling factors: 10, 20, 40, 80, 160 and 200 which corresponded to resolutions of 64×48 , 24×32 , 12×16 , 6×8 , 3×4 and 2×3 respectively. The downsampling was executed using the block-reduce function in *NumPy* which averages groups of neighbouring pixels. After that, the downsampled frames are flattened into 1d arrays which are standardized using sklearn's `StandardScaling()` transformer. Feature selection is then applied using the 'select k-best model'. Finally, the selected features along with the true labels of each frame (1 for particle A and 0 for particle B) are fed into a logistic regressor with a regularization parameter C and a liblinear solver. We sweep the regularization parameter C over values from 10^{-5} to 10^5 . A schematic of the whole pipeline is shown in Fig. 5.

More recently, we found that it is more efficient to work with histogram features generated by some modules from PROPHESEE. With these modules, we kept events with both polarities, and down-sampling was executed faster.

V. RESULTS

Single-session-training model's failed generalization: We trained the model in Fig. 5 on data from the single-session-training split model in Fig. 4(a). The result is shown in Fig. 6. The model was first tested on unseen data drawn from the same measurement session (measurement 1). This resulted in a misleadingly low error rate (below 2% as shown on the left). However, when we tested the same model on samples from other measurement sessions (sessions 2,3 and 4), we have seen that the test error rate increased significantly proving the fact that the model's weights were adjusted to learn features that are specific to the measurement and that are not attributed to the size of the particle. Moreover, the new test errors on the right shows

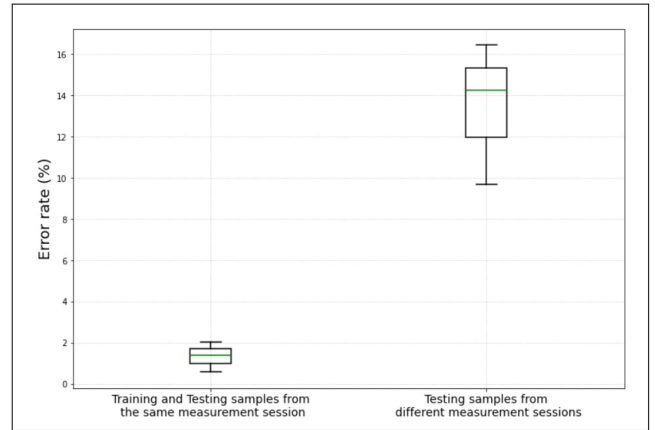


Fig. 6. Test errors of the single-session-training model. The green bars are the median of the errors over different test sets. The model achieves significantly higher accuracy when tested on data from the measurement session used in training. The lack of true generalization is revealed when other sessions are used in testing.

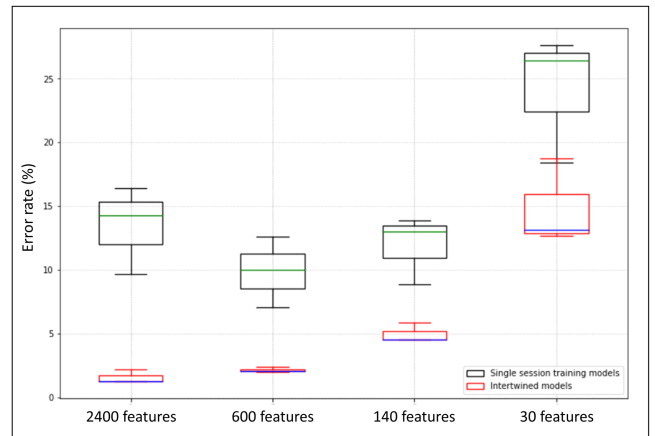


Fig. 7. Test error rates for different models trained with one single sessions (black boxes) and using samples from different measurement sessions (red boxes). Both the intertwined and single-session-training models were tested on unseen data from different sessions than those in either training or validation. On the x-axis are the resolutions of the frames sent to the linear classifier. We noticed that the intertwined models perform better than the single-session-training models at all different resolutions. In addition, the size of the red boxes are always smaller than the black boxes indicating that an intertwined model's errors on different tests are close so it is more stable than a model trained on only one single session.

relatively high variance (i.e. the box has a bigger size) meaning that the model is unstable and might perform even worse if tested on samples from a new measurement session.

Intertwined vs single session: We then took a further step and fed the same model in Fig. 5 with the intertwined data in Fig. 4(b) testing the model on sessions 2, 3 and 4. We repeated that for different downsampling factors and compared the results with the ones we got with the single-session training model. The result is shown in Fig. 7. As indicated in the figure, the test errors of the intertwined model were lower than that of the single session training model for all the different resolutions used. Additionally, the stability of the intertwined model was much better as indicated by the size of the red boxes. This gives

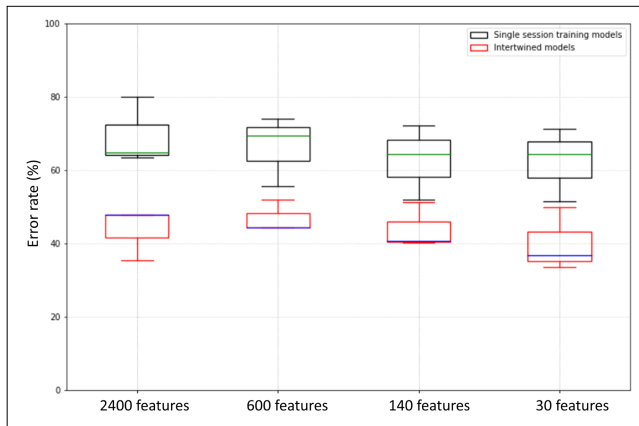


Fig. 8. Uniform mislabeling test for the intertwined model. The labels of half of the measurement sessions were mislabeled during training, then we tested the model on data from new sessions. We see that this results in error rates around 50% (with the intertwined models) which is as bad as a random guess. The single-session-training models perform better, indicating sensitivity to bias.

more confidence that the model is likely to perform similarly on samples from new measurement sessions. The highest resolution (64 x 48) resulted in a minimum error in the intertwined case, but not in the single session case which shows that additional features, which are useful for the intertwined model, cause the single-session-training model to overfit.

Note that the number of samples in the intertwined model is three times that of the single session model. Therefore, the improvement in accuracy might be attributed to the fact that we train on more samples and might have nothing to do with the fact that we eliminate measurement bias in the intertwined model. To test this hypothesis, we used a smaller data set in the intertwined models by randomly drawing only one third of the samples in each session. In the end, we could still achieve error rate below 2%, thus we conclude that the enhancement is mainly due to alleviating measurement bias by exposing the model to various sessions during training.

Compared with a CMOS-based system like the one we demonstrated in Lugnoan et al. [1], our event-based system reduced the error rate from around 10% to less than 2% while keeping the latency at a sub-millisecond value. This clearly shows that the hardware-based approach to eliminate background noise, as provided naturally by the event-based camera, is superior to a software-based technique of frame subtraction.

UM-test: An important sanity check to test the validity of our results is the so-called uniform mislabeling test. As the name suggests, in this test we mislabel some of the samples in the training dataset, then we test the trained model on unseen data. For our intertwined model, we mislabeled one of the sessions used in training by replacing the labels A1 by B1 and vice versa. This way, there is a presence of the features attributed to the particle size in both classes. Therefore, a model with mislabeled training data should not be able to classify particles based on their size. We tested our model on data from session 4. The result is shown in Fig. 8. As one can see, the model fails to classify the two particles in the test sets which is what was expected for a model that learns the

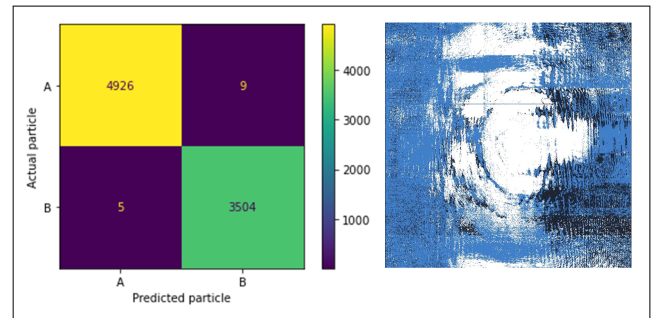


Fig. 9. Confusion matrix for one of the test sets (left) and an example of a misclassified frame (right). In this frame, Particle A is identified by the model as B. This particular error could be avoided if we apply a definite threshold on the sum of square of the pixel values to discard images with saturated signals. However, they rarely appear in our datasets.

actual data, and not the measurement conditions. On the other hand, single session training models perform better than the intertwined when trained with mislabeled training data, which shows that these models do incorporate information from the background.

Misclassified samples: From the confusion matrix of one of the test set shown in Fig. 9, we see that out of 8000 samples only 15 were misclassified. Looking at the images of the misclassified samples, we noticed that out of the 15 samples, 2 samples were saturated with events fired from most of the camera's pixels. One way to get rid of these samples is to apply a threshold on the sum of squares of pixel values. We decided not to do so because the increase in accuracy we would get is not significant, and doing so will add an extra preprocessing step which will lead to increased latency. Nevertheless, in the following section we will discuss several approaches to avoid these kind of samples in our experiment. On the other hand, we investigated the other 13 misclassified samples and could not determine, visually, why these samples were misclassified by the model.

VI. DISCUSSION

Latency and comparison with other work in the literature: In addition to the accuracy, the latency of the model is very crucial. The decision by the machine learning pipeline should not take a significantly large time, so that a cytometer would be able to receive a signal and separate the flowing cells in real time. The two main contributing steps to the latency of our model are the frame generation algorithm (which takes 0.164 ms) and the machine learning inference (which takes 0.028 ms) resulting in a 0.192 ms latency. Thanks to the event camera, we did not need to apply background subtraction on the images, which would have increased the latency. On the other hand, generating frames from the events added an extra slight delay which did not exist in the CMOS-based cytometer. Table I compares the performance of the presented system with others in literature. The main advantage of our method over most of those listed in the table stems from the fact that we use a simple linear model on features in a high dimensional space, and therefore do not require acceleration via GPU. Furthermore, not having a deep

TABLE I
COMPARING OUR WORK (LAST ROW) WITH OTHER WORK IN LITERATURE

Classification task	Classifier	Classification Performance	Accelerator	Execution time/particle
2 classes of cancer cells and 4 classes of blood cells [9]	Two nonlinear SVMs	93%	GPU	63.00 ms
Beads with diameters of 7,10 and 15 μm [27]	CNN	93% mAP	GPU	1.00 ms
3 white blood cell (WBC) types [28]	Rand. Forest on extracted features	97% accuracy	GPU	0.20 ms
1 WBC type and an epithelial cancer cell [29]	Deep CNN	96% accuracy	GPU	3.60 ms
Beads with diameters of 15.2 and 18.6 μm (CMOS work [1])	Linear (log. Regression)	90% accuracy	None	0.01 ms
Beads with diameters of 16 and 20 μm (Current work)	Linear (log. regression)	>98% accuracy	None	0.08 ms

neural network means that we can perform our training with a significantly lower amount of labelled data.

Data/memory advantage: In the previous section, we have seen that integrating an event-based camera in an imaging flow cytometer could enhance the accuracy by two orders of magnitude. Another advantage we gain is saving more memory. Ideally, our camera only stores data “events” whenever a particle passes by the field of view. This means that the size of the raw files stored are proportional to the concentration of the particles in the flowing solution. For example, we have measured a session in the beginning of our experiment with only deionized water (no particles were added). Later, we compared the size of the stored file with the average of the file size for the particle measurements. It turned out that the water measurement resulted in a file with size less than 5% of the size of a typical particle measurement of the same duration. On the other hand, a CMOS camera would result in equally large files for both measurements, since it will generate equal number of frames.

Sensitivity of the model and its importance to applications with low particle concentrations: Since the accumulation time of our event camera could be as low as 1 μs , an event-based imaging flow cytometer would have a relatively high sensitivity compared to a CMOS-based one. While in Lugnan et al. [1], many of the particles were not captured by the frame-based sensor, in this work we were able to detect fractions of particles by using a relatively low integration time. This ultra-high sensitivity is valuable for applications where a low concentration particle needs to be detected. Consider the diagnosis of circulating tumor cells (CTCs). These are cells that escape their site in the tumor to the patient’s blood. The concentration of circulating tumor cells (CTCs) in the blood could be as low as 10 cells/ml [9] requiring a flow cytometer system that has a relatively high sensitivity, which is facilitated by an event-based cytometer.

Avoiding saturations in our samples: In the previous section, examples from the misclassified particles were presented. There, it was mentioned that some of these misclassifications are due to saturated frames. We mentioned a possible preprocessing step that could remove these frames, which would however increase the system latency. Here, some alternatives are considered.

Saturated frames could be a result of a sudden change in the flow rate during measurement which leads to a decrease in the time between two consecutive particles. By making sure the flow rate is always constant during the experiment, we could avoid these situations. An automated pumping system instead of the manual syringe pump we used in the current work could guarantee this requirement.

Another way to tackle the issue is by having a dynamic instead of a static time window. The frame generation algorithm can take as an argument a fixed number of events instead of an accumulation time. By doing so, researchers managed to get rid of saturated images in their dataset [30].

Particle A is more often misclassified than particle B: By looking at the confusion matrices of different validation sets, it has been observed that particles with a smaller size (belonging to class A) are more frequently misclassified as class B (with bigger size). This could be due to the fact that clustering of smaller particles could be seen one bigger particle by the model, while this is less probable to happen with particles B.

VII. CONCLUSION

In this work, we have investigated the possibility of using event-based vision sensors in imaging flow cytometry. When doing so, we proved that both the accuracy and sensitivity of the overall system could be greatly enhanced, while keeping the latency at a relatively small value. Moreover, the prevention of measurement bias has been implemented effectively. Future work will include more automation for the pumping system which will guarantee a more stable flow rate during different measurement sessions, letting us be more confident about the estimated integration time used to generate cell frames from events. Furthermore, in contrast to our supervised training model, semi-supervised methods based on graph signal processing could be suitable for this application due to the fact that they require less labeled data than supervised methods and are more robust than unsupervised methods [31], [32], [33]. Finally, since most publications on flow cytometry use biological cells instead of microparticles as classes, testing our system with cells will allow us to objectively compare it with those in literature.

REFERENCES

- [1] A. Lugnan, E. Gooskens, J. Vatin, J. Dambre, and P. Bienstman, "Machine learning issues and opportunities in ultrafast particle classification for label-free microflow cytometry," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, 2020.
- [2] M. Doan et al., "Diagnostic potential of imaging flow cytometry," *Trends Biotechnol.*, vol. 36, no. 7, pp. 649–652, 2018.
- [3] Y. Li et al., "Deep cytometry: Deep learning with real-time inference in cell sorting and flow cytometry," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019.
- [4] N. Nitta et al., "Intelligent image-activated cell sorting," *Cell*, vol. 175, no. 1, pp. 266–276, 2018.
- [5] T. C. George et al., "Distinguishing modes of cell death using the ImageStream multispectral imaging flow cytometer," *Cytometry Part A: J. Int. Soc. Anal. Cytol.*, vol. 59, no. 2, pp. 237–245, 2004.
- [6] Y. Han, Y. Gu, A. C. Zhang, and Y.-H. Lo, "Imaging technologies for flow cytometry," *Lab Chip*, vol. 16, no. 24, pp. 4639–4647, 2016.
- [7] M. Hussain et al., "On-chip classification of micro-particles using laser light scattering and machine learning," *Chin. Chem. Lett.*, vol. 22, pp. 1885–1888, 2021.
- [8] Z. Zhuo, H. Wang, R. Liao, and H. Ma, "Machine learning powered microalgae classification by use of polarized light scattering data," *Appl. Sci.*, vol. 12, no. 7, 2022, Art. no. 3422.
- [9] N. Nissim, M. Dudaie, I. Barnea, and N. T. Shaked, "Real-time stain-free classification of cancer cells and blood cells using interferometric phase microscopy and machine learning," *Cytometry Part A*, vol. 99, no. 5, pp. 511–523, 2021.
- [10] S. Ota, I. Sato, and R. Horisaki, "Implementing machine learning methods for imaging flow cytometry," *Microscopy*, vol. 69, no. 2, pp. 61–68, 2020.
- [11] R. Vaghashiya et al., "Machine learning based lens-free shadow imaging technique for field-portable cytometry," *Biosensors*, vol. 12, no. 3, 2022, Art. no. 144.
- [12] R. Geirhos et al., "Shortcut learning in deep neural networks," *Nature Mach. Intell.*, vol. 2, no. 11, pp. 665–673, 2020.
- [13] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorph event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct. 2014.
- [14] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15 μ s latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
- [15] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16639–16652.
- [16] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5419–5427.
- [17] S. Barua, Y. Miyatani, and A. Veeraraghavan, "Direct face detection and video reconstruction from event cameras," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–9.
- [18] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.
- [19] K. Golibrzuch, S. Schwabe, T. Zhong, K. Papendorf, and A. M. Wodtke, "Application of an event-based camera for real-time velocity resolved kinetics," *J. Phys. Chem. A*, vol. 126, pp. 2142–214, 2022.
- [20] A. Amir et al., "A low power, fully event-based gesture recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7243–7252.
- [21] A. Sokolova and A. Konushin, "Human identification by gait from event-based camera," in *Proc. 16th Int. Conf. Mach. Vis. Appl.*, 2019, pp. 1–6.
- [22] S. Schaefer, D. Gehrig, and D. Scaramuzza, "AEGNN: Asynchronous event-based graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12371–12381.
- [23] A. Mondal et al., "Moving object detection for event-based vision using graph spectral clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 876–884.
- [24] S. Zhang, W. Wang, H. Li, and S. Zhang, "EventMD: High-speed moving object detection based on event-based video frames," 2022. [Online]. Available: <https://ssrn.com/abstract=4006876>
- [25] J. Zhao, S. Ji, Z. Cai, Y. Zeng, and Y. Wang, "Moving object detection and tracking by event frame from neuromorphic vision sensors," *Biomimetics*, vol. 7, no. 1, 2022, Art. no. 31.
- [26] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
- [27] Y. J. Heo, D. Lee, J. Kang, K. Lee, and W. K. Chung, "Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip," *Sci. Rep.*, vol. 7, no. 1, pp. 1–9, 2017.
- [28] B. Cornelis, D. Blinder, B. Jansen, L. Lagae, and P. Schelkens, "Fast and robust fourier domain-based classification for on-chip lens-free flow cytometry," *Opt. Exp.*, vol. 26, no. 11, pp. 14329–14339, 2018.
- [29] K. Goda and B. Jalali, "Dispersive fourier transformation for fast continuous single-shot measurements," *Nature Photon.*, vol. 7, no. 2, pp. 102–112, 2013.
- [30] R. Ghosh, A. Mishra, G. Orchard, and N. V. Thakor, "Real-time object recognition and orientation estimation using an event-based camera and cnn," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2014, pp. 544–547.
- [31] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vanderghenst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [32] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, Nov. 2020.
- [33] J. H. Giraldo and T. Bouwmans, "Semi-supervised background subtraction of unseen videos: Minimization of the total variation of graph signals," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3224–3228.

Muhammed Gouda was born in Gharbeya, Egypt, in 1995. He received the bachelor's degree in nanotechnology and nanoelectronics engineering from the Zewail City of Science and Technology, Egypt, in 2018, two years later, the master's degree in photonics from Ghent University, Ghent, Belgium, and Free University of Brussels, Brussels, Belgium. In 2020, he joined the Photonics Research Group as a Ph.D. student to work on neuromorphic computing using programmable photonics.

Alessio Lugnan received the master's degree in experimental physics from the University of Trento, Trento, Italy, in 2016, and the Ph.D. degree in photonics engineering in 2021. He joined the Photonics Research Group, Ghent University – Imec, Ghent, Belgium, in 2016, as a Ph.D. student, working on optical solutions for machine learning classification of particles for imaging microflow cytometry and integrated photonic reservoir computing using ring resonators. He is currently a Postdoctoral Researcher with the same group, on neuromorphic computing with silicon photonics and phase change materials.

Joni Dambre received the M.Sc. degree in electrical engineering and the Ph.D. degree in computer science engineering from Ghent University, Ghent, Belgium. After doing his Ph.D. and Postdoc research in topics related to digital hardware (circuits design, design automation), she made a successful career shift toward machine learning and neural networks in 2008, obtaining a 10% professorship (UGent). In 2013, she became a Full-time Tenured Professor (hoofddocent). Her Research Group (now called AIRO) covers a broad spectrum of interconnected topics related to machine learning, deep learning and embedded implementations thereof, robotics and brain-inspired unconventional computing. Under her lead, the team combines foundational and applied research. A thorough hands-on applied knowledge is being maintained through the team's successful track record in Kaggle competitions.

Gerd van den Branden received the master's degree in electronics from the KULeuven (campus De Nayer), Leuven, Belgium, in 2003. He started his career with the Erasmushogeschool Brussel as a Scientific Researcher on dynamic reconfiguration in FPGA technology. In 2004, he joined the ETRO Research Group with the Vrije Universiteit Brussel (University of Brussels), Brussels, Belgium. In 2007, he joined the CMOS Image Sensor Group of Cypress Semiconductor as a Test Engineer. From 2011 onwards, he was leading the System Development Group of the Image Sensor Group, ON Semiconductor (currently known as Onsemi). As part of the Global R&D Organization, he contributed to architecture, develop and deploy the next generation image sensor validation, demo and production test platforms worldwide. In 2017, he joined Prophesee S.A. as the Director of System and Industrialization, building up from the ground the organization that industrializes and qualifies event-based technology for mass-production. In 2020, he became a Co-founder with a Belgian based company named Industrialise B.V., which provides engineering design services and IP to semiconductor and electronic module industry.

Peter Bienstman (Member, IEE) was born in Ghent, Belgium, in 1974. He received a degree in electrical engineering from Ghent University, Ghent, Belgium, in 1997, and the Ph.D. degree from the Department of Information Technology (INTEC), Ghent University, in 2001. He is currently a Full Professor with INTEC. He has authored or coauthored more than 110 papers and holds several patents. His research interests include the several applications of nanophotonics (biosensors and photonic information processing,) and nanophotonics modeling. He was the recipient of a ERC starting grant for the Naresco-Project: Novel paradigms for massively parallel nanophotonic information processing.

Christoph Posch received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Technology, Vienna, Austria, in 1995 and 1999, respectively. After working on CMOS IC design for high-energy physics experiments with CERN from 1996 to 2004, he helped founding the Smart Sensors Group, Austrian Institute of Technology, where he became Principal Scientist in 2007. In 2012, he Co-directed the Neuromorphic Vision Group, Institut de la Vision, Paris, France, and was a Research Professor with Sorbonne Universit  Paris VI. He is a Co-founder and the CTO of Prophesee, Paris, France, a start-up company developing Neuromorphic event-based vision technology. He has authored more than 100 scientific publications and holds more than 15 patents in the fields of artificial vision sensing and processing. Dr. Posch was the recipient of eight IEEE awards.