Motion Enhanced Multi-Level Tracker (MEMTrack): A Deep Learning-Based Approach to Microrobot Tracking in Dense and Low-Contrast Environments

Medha Sawhney, Bhas Karmarkar, Eric J. Leaman, Arka Daw, Anuj Karpatne, and Bahareh Behkam**

Tracking microrobots is challenging due to their minute size and high speed. In biomedical applications, this challenge is exacerbated by the dense surrounding environments with feature sizes and shapes comparable to microrobots. Herein, Motion Enhanced Multi-level Tracker (MEMTrack) is introduced for detecting and tracking microrobots in dense and low-contrast environments. Informed by the physics of microrobot motion, synthetic motion features for deep learning-based object detection and a modified Simple Online and Real-time Tracking (SORT) algorithm with interpolation are used for tracking. MEMTrack is trained and tested using bacterial micromotors in collagen (tissue phantom), achieving precision and recall of 76% and 51%, respectively. Compared to the state-of-the-art baseline models, MEMTrack provides a minimum of 2.6-fold higher precision with a reasonably high recall. MEMTrack's generalizability to unseen (aqueous) media and its versatility in tracking microrobots of different shapes, sizes, and motion characteristics are shown. Finally, it is shown that MEMTrack localizes objects with a root-mean-square error of less than 1.84 μm and quantifies the average speed of all tested systems with no statistically significant difference from the laboriously produced manual tracking data. MEMTrack significantly advances microrobot localization and tracking in dense and low-contrast settings and can impact fundamental and translational microrobotic research.

1. Introduction

Microrobotic systems continue to garner significant interest due to their potential in various fields, including targeted drug delivery, minimally invasive surgery, and biosensing.[1] Based on their mode of actuation and target application, microrobots range between ≈1 and 1000 μm in size, and their average speeds vary vastly from 1 to 800 $\mu m\ s^{-1}$ with instantaneous speeds upward of 800 $\mu m\ s^{-1}$.[2,3] These properties make microrobots very effective in reaching currently inaccessible areas of the human body but incredibly difficult to visualize and track. Traditionally, microrobots have been studied in aqueous environments (**Figure 1**A–B). The growing focus shift in the microrobotic field from system development to biomedical application-oriented implementations necessitates operating and controlling such systems in physiologically relevant environments, and aqueous media do not always represent the conditions and interactions experienced in vivo. Furthermore, advancing the current understanding of the physical underpinnings of microrobot behaviors in vivo requires studying these systems in physiologically relevant in vitro, ex vivo, or in vivo environments.[4]

The fast speeds of microrobotic systems (i.e., 10s-100s of body lengths per second) necessitate high frame rate image acquisition, primarily attainable using bright-field imaging. The resulting grayscale images (Figure 1B) have significantly lower contrast than fluorescent images (Figure 1A), typically used for automated localization and tracking. Also, microrobots (≈1 μm in size) often swim in 3D space, which translates to intermittent movement of the objects of interest in and out of the focal plane, adding another layer of complexity. Moreover, self-propelled microrobots (e.g., biohybrids or catalytic motors) exhibit random walk behavior, making it difficult to track them consistently in every frame. Operation in physiologically relevant environments

M. Sawhney, A. Daw, A. Karpatne
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061, USA
E-mail: karpatne@vt.edu

B. Karmarkar, E. J. Leaman, B. Behkam
Department of Mechanical Engineering
Virginia Tech
Blacksburg, VA 24061, USA
E-mail: behkam@vt.edu

B. Behkam
School of Biomedical Engineering & Sciences
Macromolecules Innovation Institute
Center for Engineered Health
Center for Soft Matter and Biological Physics
Virginia Tech
Blacksburg, VA 24061, USA

DOI: 10.1002/aisy.202300590

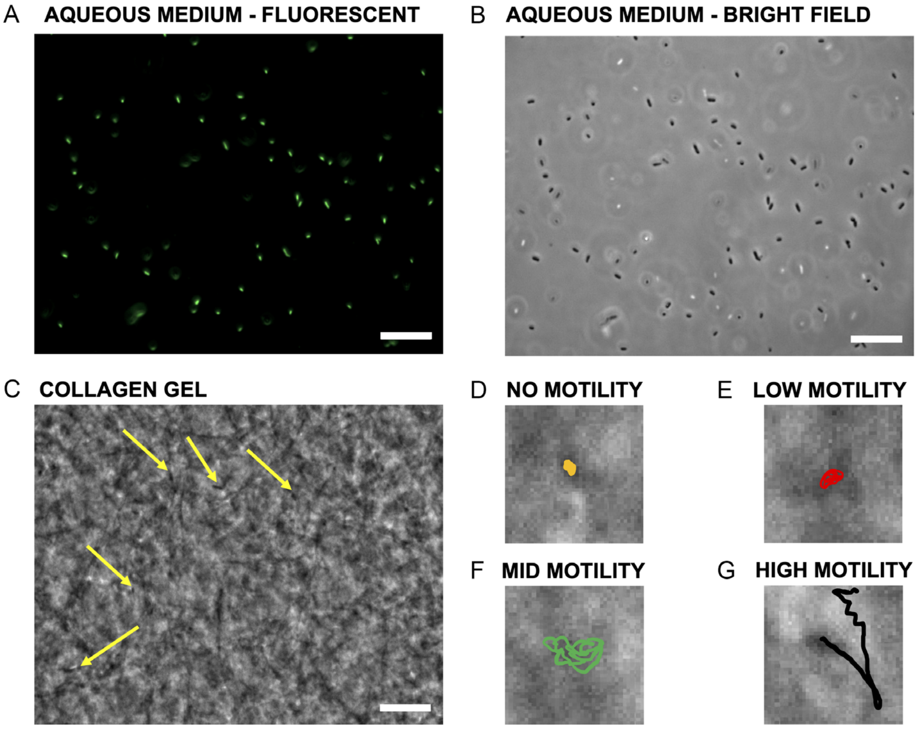

Figure 1. A) Microrobots, including bacterial biohybrid microrobots, are easy to track in high-contrast fluorescent images; B) however, they are more difficult to detect in high acquisition rate bright-field image sequences in liquid (aqueous) medium. C) The complexity further increases in dense environments such as collagen with feature shapes and sizes similar to those of microrobots. The yellow arrows represent microrobots' locations. D–G) Sample trajectories of bacteria in different motility subpopulations. All scale bars are 20 μm.

with dense backgrounds and feature sizes and shapes comparable to those of microrobots (Figure 1C) further exacerbates these problems.

There has been significant progress in real-time tracking of microrobots using clinical imaging modalities that produce high-contrast images;[5–11] however, automated tracking and localization of microrobots in bright-field image sequences within dense backgrounds remains a largely unsolved challenge. A few commonly used techniques exist for manual or semiautomated tracking of microscale objects. Each technique has different features and capabilities, such as particle identification and tracking or morphology analysis. The most widely used tool for manual tracking is Fiji (ImageJ2), an open-source software for the processing and analysis of scientific images.[12] Fiji is equipped with several plugins for manual and semiautomated tracking, including Trackmate[13,14], MtrackJ[15], CellProfiler[16,17], MosaicSuite-Particle Tracker[18], FARSIGHT[19], BioImageXD[20], and Icy[21]. The semiautomated methods require user inputs, such as filters or thresholds and manual intervention to prune false positive (FP) data. None of these methods were explicitly developed for object detection in dense backgrounds, where the visual ambiguity between the dense background and microscale objects makes tracking error-prone, tedious, and time-consuming.

Apart from these commonly used tools, several other methods have been developed for cells, nanoparticles, and small object detection and tracking in 2D and 3D spaces.[22–33] However, most methods primarily rely on the high contrast between the object of interest and the background that is unique to fluorescent images. The longer exposure time of fluorescence imaging (≈100s ms compared to ≈10s ms for bright-field imaging) results in a reduced acquisition rate and loss of important temporal data, making it unfeasible for many microrobotic applications. Several models have also been developed for tracking objects in aqueous environments using bright-field images. These models consider cell appearance changes and overlaps during colony proliferation in time-lapse videos.[34,35] They can also track single cells in aqueous media but are still not fully automated and require manual interventions.[30,36–41] To the authors' best knowledge, automated tracking of microrobots in dense and low-contrast environments, with features and dimensions similar to those of objects of interest, has not yet been realized.

In this work, we report the development of a deep learning-based approach to address the gap in the detection and tracking of microrobots in dense environments. Existing deep learning-based multiobject tracking works have primarily focused on detecting and tracking objects easily distinguished from the background (e.g., pedestrians and cars).[42] In this work, we present the Motion Enhanced Multi-level Tracker (MEMTrack) for tracking microscale objects in dense environments, where the object of interest is almost indistinguishable from the background features (Figure 1C). We train and test MEMTrack using bacterial micro-biomotors, one of the most commonly used biomotors in biohybrid microrobotic systems.[43] To illustrate the broad utility of the MEMTrack in various backgrounds, we demonstrate its performance in collagen, the most abundant extracellular matrix protein in the body, with feature sizes and shapes comparable to those of bacteria (Figure 1C) and in aqueous media (Figure 1B). To illustrate MEMTrack's versatility, we demonstrate its performance in tracking microrobots of different shapes, sizes, and motion characteristics. Our results demonstrate that our pipeline can accurately predict and track both visually identifiable and hard-to-detect microrobots. Our proposed pipeline represents a significant contribution to localization and tracking in dense and low-contrast settings, which can potentially impact both fundamental and translational microrobotic research. Moreover, it opens the potential of applying deep learning-based methods for vision-based control of microrobots in dense and low-contrast settings for various applications, including disease diagnosis and treatment.

2. Methods

2.1. MEMTrack

The proposed pipeline for MEMTrack is shown in **Figure 2**. MEMTrack consists of four modules—Motion Enhancer, Multi-level Object Detector, False Positive Pruner, and Interpolated Tracker. Before describing each module, we define the notations used throughout the article. We define the input video with T frames as $I^{1\ldots\mathrm{T}} = [I^1, I^2, \ldots, I^\mathrm{T}]$, where $I^t \in \mathbb{R}^{\mathrm{C}\times\mathrm{H}\times\mathrm{W}}$. denotes the t^{th} frame and C, H, and W are the number of channels, height, and width of the image, respectively. We utilized a tracking-by-detection approach, wherein the tracking is done on top of predictions from the detection module.

2.1.1. Motion Enhancer Module

Object detectors, designed for single-frame detection, overlook the object's position in preceding or subsequent frames, limiting their tracking effectiveness. Incorporating the concept of motion into the object detection model is crucial to enable accurate detection of microscale objects in dense environments with intermixing backgrounds and foregrounds (Figure 1C). Therefore, we implemented feature engineering techniques to capture two types of motion features, optical flow features and median deviation features, which are then augmented or stacked with the image features to enhance detection accuracy, as shown in Figure 2A and described below.

Optical Flow Features: Optical flow[44] is a technique that estimates the motion of objects in an image sequence by analyzing the changes in pixel intensities between consecutive frames. We used the Lucas–Kanade[45] method for optical flow computation, which is expressed mathematically as

$$\partial I_x u + \partial I_y v + \partial I_t = 0 \tag{1}$$

where $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ represent the x and y components of the optical flow vector for the t^{th} frame, and, $\partial I_x = \frac{dI}{dx}$, $\partial I_y = \frac{dI}{dy}$, and $\partial I_t = \frac{dI}{dt}$ are the image gradients in the x, y, and time (t) dimensions, respectively. Solving this equation yields the optical flow vector $O = [u, v]$ for the t^{th} frame in the video. The length of this optical flow vector, which corresponds to the magnitude of motion at each pixel between consecutive frames, is then considered as an additional feature channel along with the grayscale image.

Median Deviation Features: While optical flow captures changes across consecutive frames, we also need features that capture slower motion trends with respect to a static background. To this effect, we used the median deviation as another feature channel, which is the pixel-wise difference between the intensity at a pixel and the median intensity at the pixel across all frames in

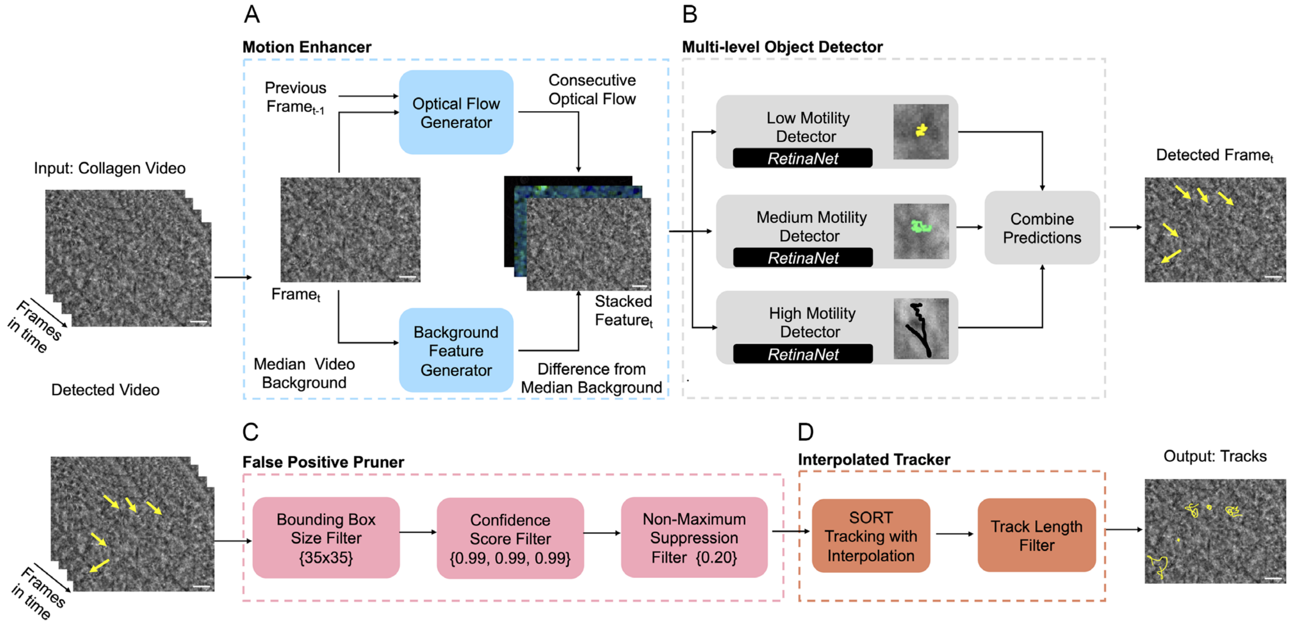

Figure 2. Overview of the proposed MEMTrack pipeline consisting of four modules: A) Motion Enhancer, which adds motion features to the input frames, B) Multi-level Object Detector, which detects objects of varying motility levels using the deep learning-based RetinaNet model, C) False Positive Pruner, which filters the predicted object occurrences to reduce false positives, and D) Interpolated Tracker, which tracks objects over time using the SORT algorithm with linear interpolation.

the video. Note that the median represents the background at every pixel that remains static for the majority of frames. Mathematically, we defined median deviation $\Delta I = |I^{t} - \mathrm{median}(I^{1\cdots T})|$, where median $(I^{1\ldots T})$ is the pixel-wise median for the video across frames 1 through T.

Finally, the input image features I^{t}, concatenated with the optical flow features O and the median deviation ΔI, are used as inputs to the Multi-level Object Detector.

2.1.2. Multi-Level Object Detector Module

Object detection is a fundamental task in computer vision and plays a crucial role in various applications such as automation and decision-making. The two main approaches used are two-stage detectors and one-stage detectors. Two-stage detectors such as Region-based Convolutional Neural Network (R-CNN)[46] and Fast R-CNN[47] offer high accuracy but suffer from high computational complexity, while one-stage detectors such as You Only Look Once (YOLO)[48–51] provide faster inference but may struggle with detection of small objects in dense environments reference.[52–56]

In our pipeline, we used the one-stage deep learning detector RetinaNet[57] as the base object detector. RetinaNet introduces the novel "focal loss" function, which assigns higher weights to frequently misclassified objects during training, improving accuracy without sacrificing speed. RetinaNet strikes a balance between accuracy and efficiency, making it suitable for real-time tracking in complex backgrounds, as required for microrobot tracking in dense environments. As is the case with most microrobots, bacterial biohybrid microrobots exhibit various motion patterns and speeds (Figures 1D–G and Video S1, Supporting Information) with different detection requirements, making a single-model training for all the observed behaviors ineffective. Informed by the physics of the microrobot motion, we propose a Multi-level Object Detection model, where we train a different detector model for each motility category of low, medium, and high (Figure 2B, Section 2.2). As each detector model is only exposed to the ground truth (GT) data of a specific motility category during training, its learning is tailored to capture the specific motion characteristics of bacteria belonging to a single category. This approach enhances the accuracy of the object detection system for each category and improves the overall performance of the system, as discussed in Section 3.

During the training phase, the object detector receives input from annotated microscopy videos with bounding boxes of a prescribed size centered around microrobots' centroids. When performing inference, the object detectors predict coordinates (x, y) for bounding boxes to indicate the presence of objects and their width and height. These predictions are accompanied by a confidence score reflecting the model's certainty level regarding the detection.

2.1.3. False Positive Pruner Module

The Multi-level Object Detector module can produce a large number of as well as duplicate predictions from the three different detector models. The False Positive Pruner was implemented to remove the FPs without losing the true positive (TP) predictions. To this end, first, we combined all the detections from the three motility models and then pruned the detections based on three exclusionary criteria, as shown in Figure 2C: 1) Bounding Box Filter for removal of predictions that are greater than a prescribed area threshold, 2) Confidence Score Filter for removal of predictions lower than a prescribed confidence threshold, and 3) Non-maximum Suppression (NMS)[58] Filter for elimination of redundant object detections by selecting the ones with the highest confidence score and discarding the other overlapping ones. NMS evaluates the Intersection over Union (IoU = Area of Intersection/Area of Union) between detected bounding boxes. The threshold on IoU serves as a basis to determine whether detected boxes correspond to the same object in the NMS Filter. Beginning with the most confident detected box, NMS eliminates overlapping boxes with lower confidence scores, resulting in accurate object selection and reduced redundancy of detections. Section 2.3.4 describes our process for selecting the thresholds for the three filters.

2.1.4. Interpolated Tracker Module

Several approaches have been proposed for object tracking, including correlation filter-based[58], Kalman filter-based[59], and deep learning-based[60,61] methods. Simple Online Real-time Tracking (SORT)[59] is one of the most widely used tracking algorithms. It uses a combination of Kalman filtering[62] and the Hungarian algorithm[63,64] to assign detected objects to existing tracks. The Kalman filter in SORT works by recursively updating estimates of the current system state (in our case, positions of microrobots) based on the previous configuration of the state and the current measurements (i.e., the detected microrobot positions in the current frame) while also taking into account the uncertainty of those measurements. SORT is simple and efficient and performs well on various tracking tasks, such as pedestrian or vehicle tracking. However, the random motion of microrobots and intermittent missing detections resulting from their 3D motion may limit the performance of this algorithm.

Our proposed method modifies the SORT algorithm by incorporating linear interpolations for missed object detections. As shown in Figure 2D, we applied the SORT algorithm to track the detected microrobots and produce tracklets from independent frame-wise predictions. We interpolated the missing detections by keeping the Kalman filter-based unmatched predictions for a given number of frames, termed the maximum age parameter (determined in Section 2.3.4), and dropping the predictions beyond that threshold. The maximum age parameter within the SORT algorithm ensures the persistence of a tracklet for a specified number of frames subsequent to a missed detection event, thereby upholding tracking continuity. If a track is missed further than the maximum age, it is discarded.

Finally, we introduce the Track Length Filter to remove tracks whose length, in terms of the number of frames, does not meet a specified threshold (determined in Section 2.3.4). This filter helps exclude excessively short tracks that are prone to being FPs.

2.2. Experimental Methods

In order to evaluate the performance of MEMTrack, we recorded bacteria (i.e., the micromotors in bacterial biohybrid microrobots) swimming behavior in collagen, as a tissue surrogate, and in an aqueous environment, as a significantly different unseen medium.[65,66]

2.2.1. Bacteria Culture

Six engineered strains of *Salmonella* Typhimurium VNP20009cheY^{+}[67] bacteria with different motile behaviors were used. Each strain was grown on a 1.5% lysogeny broth (LB; 1% tryptone, 0.5% yeast extract, and 1% sodium chloride) agar plate overnight at 37 °C. For each experiment, a single colony of the desired strain was isolated and used to inoculate 10 mL of LB media in a 125 mL smooth-bottom flask. Bacteria were cultured overnight at 37 °C and 100 RPM before being harvested and resuspended in fresh LB to a final concentration of $\approx 1.3 \times 10^{6}$ CFU mL^{-1}.

2.2.2. Swimming Assay in Collagen and Aqueous Medium

Bacteria motility in collagen was evaluated using an experimental setup similar to the traditional swim plate assay, in which bacteria migrate outward from a central inoculation point due to a combination of chemotaxis and growth. Collagen gel was prepared from a stock solution of collagen type I that was neutralized with 0.25 N NaOH, diluted to 5 mg mL^{-1} in LB, and supplemented with 100 μg mL^{-1} ampicillin on ice. The cold collagen solution was pipetted into wells of a room-temperature well plate. The well plate was then immediately placed in a 37 °C incubator for 45 min to allow the collagen to gel. A 1 μL aliquot of bacterial suspension, prepared as described in Section 2.2.1, was introduced at the center of each collagen well. All data acquisition was performed using a Zeiss AxioObserver.Z1 inverted microscope equipped with a 40× objective and an hSM camera (Carl Zeiss AG, Oberkochen, Germany). Phase contrast microscopy images were collected at 60 frames per second (FPS) for 2 min at 37 °C.

For motility assays in aqueous media, a 100 μL aliquot of the overnight bacteria culture, described in 2.2.1, was used to inoculate a fresh LB culture. The culture was grown at 37 °C for ≈2.5 h or until $OD_{600} = 1$was reached. Subsequently, the culture was diluted 20× in fresh LB medium. A 10 μL aliquot of the bacterial suspension was placed between two No. 1.5 coverslips separated by a thin ring of vacuum grease. Time-lapse imaging was performed as described above at 60 FPS for 10 s. For MEMTrack comparison with baseline models, the aqueous swimming assays were performed according to our previously developed methods.[65]

2.2.3. Bacteria Tracking and Annotation

In order to generate training, validation, and test datasets for MEMTrack, the microscopy videos acquired in collagen and in aqueous media were imported into ImageJ[12] software. MTrackJ[15] plugin was used to manually label all bacteria in each video frame (≈10–20 bacteria per frame), and their x and y coordinates were recorded.

2.2.4. Determining Track Length and Motion Characteristics

In order to determine the tracking period threshold for capturing bacteria random walk in collagen, we first tracked the bacteria for 150 frames and evaluated the diffusivity of each bacterium according to

$$D(\tau) = \sum_{0}^{2.5} \frac{(x(t+\tau) - x_0(t))2 + (y(t+\tau) - y_0(t))2}{4\tau} \tag{2}$$

where $\tau = 0.016$ s is the lag time between consecutive frames, and x_0 and y_0 represent the initial position. As shown in Figure S1, Supporting Information, the diffusivity values plateaued at or before 1 s, indicating that a track length of 1 s (60 frames) in collagen is sufficient for capturing bacteria's random walk.

Next, we used peak diffusivity values to divide bacteria into four categories based on their motility patterns and the associated diffusivity value—no motility ($D_{\text{peak}} \leq 0.075\ \mu\text{m}^2\ \text{s}^{-1}$), low motility ($0.075\ \mu\text{m}^2\ \text{s}^{-1} < D_{\text{peak}} \leq 0.25\ \mu\text{m}^2\ \text{s}^{-1}$), medium motility ($0.25\ \mu\text{m}^2\ \text{s}^{-1} < D_{\text{peak}} \leq 1\ \mu\text{m}^2\ \text{s}^{-1}$), and high motility ($D > 1\ \mu\text{m}^2\ \text{s}^{-1}$). Representative trajectories for each subpopulation are shown in Figure S2, Supporting Information.

Based on the well-known motile behavior of bacteria in aqueous environments,[68] we selected 0.5 s (30 frames) for testing MEMTrack's performance on aqueous media datasets.

2.3. In Silico Experimental Setup

2.3.1. Training, Validation, and Testing Datasets

MEMTrack was trained and validated using the experimental datasets in collagen, while its performance was evaluated using both the collagen and the aqueous medium datasets. A subset of the collagen data, termed the training set, was used for learning the model parameters (i.e., weights and biases of the deep learning models) using gradient descent algorithms. Simultaneously, another subset of the collagen data, termed the validation set, was used during training for observing the performance of the model on data outside the training set and determining the configuration of hyperparameters (i.e., parameters of the model that are not directly learned using gradient descent such as the filter thresholds of the False Positive Pruner module) that yields best validation performance. Finally, a third subset of the collagen data, termed the test set, which has no overlap with the other two sets, was used to report the model performance on "unseen" data. We used the entirety of the aqueous medium data for testing the collagen-trained model in that medium.

To ensure consistency and provide comprehensive training data for different motion types and background scenarios across the six bacterial strains in the collagen, we allocated two videos per strain for training (a total of twelve) and reserved one video per strain for validation (a total of six). The collagen test set consisted of 16 videos representing all bacterial strains. The aqueous medium dataset contained five videos that were all used for testing.

2.3.2. Evaluation Metrics

We used precision (Equation (3)), recall (Equation (4)), mean Intersection over Union (mIoU, Equation (5)), and root mean square error (RMSE, Equation (6)) to evaluate the performance of MEMTrack quantitatively. Precision is the fraction of the TP predictions over the sum of all predictions (TPs and FPs). It is a measure of how precise or confident the model is in tracking real objects (e.g., bacteria)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

Recall, on the other hand, signifies the fraction of GT or actual objects (e.g., bacteria) that the model is able to recover. It is defined as the fraction of the TP over the GT (i.e., the summation of TP and false negatives (FNs))

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

IoU compares the predicted and GT bounding boxes and is defined as

$$\text{IoU} = \frac{\text{Area of intersection}}{\text{Area of Union}} \tag{5}$$

The mIoU is calculated by summing the IoU values for all pairs of GT and corresponding predicted TP matches, and dividing this total by the number of matching pairs.

RMSE is a measure of the difference between the GT and the predicted TP coordinates, and is calculated using

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}((x_{\text{GT},i} - x_{\text{pred},i})^2 + (y_{\text{GT},i} - y_{\text{pred},i})^2)}{N}} \tag{6}$$

Here, $x_{\text{GT},i}$ and $y_{\text{GT},i}$ are the GT coordinates and $x_{\text{pred},i}$ and $y_{\text{pred},i}$ are the predicted TP coordinates. We calculated the RMSE over all TPs.

Finally, we use the F1 score[69], the harmonic mean of precision and recall, as another evaluation metric to select hyperparameters in our model architecture.

$$\text{F1 Score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

2.3.3. Baseline Models

To evaluate MEMTrack's performance, we selected four baseline methods to cover a comprehensive range of detection techniques—classical computer vision, background detection, and deep learning—and tracking methods, including filter-based and optimization-based approaches like Linear Assignment Problem (LAP). Below, we briefly describe the categories of the baseline methods.

Classical Computer Vision-Based Methods: Trackmate7[13,14] and MosaicSuite-Particle Tracker[18] fall into this category. Trackmate7 implements a Laplacian of Gaussian (LoG)[70,71]-like detection approach, augmented with thresholding, and employs a LAP tracker[72,73]. This optimization-based tracker effectively assigns objects to tracks based on similarity metrics. MosaicSuite, while also using LoG-based detection, differentiates itself with a particle filter-based feature point tracking algorithm. This algorithm is self-initializing and skilled in filtering out spurious detections, demonstrating its efficacy in complex tracking scenarios.

Background Detection-Based Methods: Your Software for Motility Recognition (YSMR)[36] integrates Gaussian blur with adaptive thresholding for object detection, followed by centroid tracking. This method efficiently tracks object movements by focusing on geometric centers, proving particularly useful in environments with variable background conditions. Notably, YSMR requires careful tuning of over 30 parameters to optimally adapt to the specific data being used, which adds to its adaptability but also its complexity.

Deep Learning-Based Methods: DEtecion TRansformer (DETR)[74] harnesses the capabilities of transformers[75] for object detection. Tracking is achieved using SORT[61], a filter-based method applied to the detections from DETR. SORT employs a Kalman filter with a linear assumption, contrasting the particle filter approach in its tracking methodology.

The selected hyperparameters and model training specifics, when applicable, used to evaluate each baseline model are described in Section S.A, Supporting Information.

2.3.4. Determining the Model Hyperparameters

Model hyperparameters include the configurational parameters of a machine learning model that are not trained directly using gradient descent but need to be set manually, informed by domain knowledge, before the training begins. The hyperparameters of the MEMTrack platform are the area threshold (Bounding Box Filter), confidence thresholds (Confidence Score Filter), IoU threshold (NMS Filter), maximum age parameter (Tracking Interpolation), and track length threshold (Track Length Filter). The selection of these parameters is guided either by domain-specific insights relevant to microrobot motion characteristics or through a grid search process, ensuring optimal performance.

The area threshold is tailored to the average diameter of the object of interest, ensuring predicted bounding boxes align with expected object sizes. In the case of bacterial micromotors, the area threshold in the Bounding Box Filter was set to a value of 35 × 35 pixels ($8.68 \times 8.68\,\mu\text{m}^2$). Consequently, any detected object characterized by a bounding box area exceeding 35 × 35 pixels was omitted from our predictions. Note that this threshold exceeds the training bounding box size criterion of 30 × 30 pixels ($7.44 \times 7.44\,\mu\text{m}^2$), based on average and maximum bacteria sizes of $3.4 \pm 0.7\,\mu\text{m}$ and $\approx 5\,\mu\text{m}$, respectively, to be tolerant of predictions with larger bounding box size during inference compared to training.

We experimented with two possible criteria for selecting the Confidence Score Filter thresholds of the three-level object detector using the validation set—maximum precision criterion and maximum F1 score criterion. When the maximum precision criterion is used, the three Confidence Score thresholds (one for each motility level) are set to maximize the precision on the validation set for each motility level. This criterion ensures that the number of FPs is minimized in our predictions across all motility levels. However, by solely maximizing the precision, we cannot control for the number of FNs and hence may obtain lower recall values. On the other hand, the maximum F1 score criterion ensures that the harmonic mean of precision and recall values are high for all three motility models, thus making a trade-off between reducing FPs and improving recall of the GT bacteria. The choice of maximum precision versus maximum F1 score criteria depends on the user's priorities. While the maximum F1 score criterion would yield a higher recall than the maximum precision criterion, it would do so at the cost of a lower precision value (or, equivalently, more FP tracks), requiring additional manual intervention to remove the FP tracks. **Figure 3** shows the precision–recall curves for the validation set of the three motility levels of the Multi-level Object Detector, where the threshold settings for the two criteria are indicated as vertical lines. For the remainder of this paper, all results from the MEMTrack model were obtained using the maximum precision criterion unless otherwise indicated, because our focus was to obtain predictions of bacteria tracks with high precision without requiring additional manual postprocessing to remove FP tracks.

For optimizing the IoU threshold (NMS Filter) and the maximum age parameter (SORT Tracking with Interpolation), we conducted comprehensive grid searches. The range for the IoU threshold was set between 0 and 1, with a step size of 0.1, while the maximum age parameter was varied from 0 to 100 at increments of 5, considering our typical video length of 100-150 frames. We optimized the grid search to maximize the F1 score, as the interpolated tracking in SORT is designed to compensate for any fragments in the predicted tracks. The IoU threshold is set to 0.2. The maximum age threshold for the SORT Tracking was set to 35 frames, indicating that interpolation for absent detections is conducted for up to 35 missed frames.

For the Track Length Filter, we used a threshold of 60 minimum frames to capture bacteria random walk in collagen, as described in Section 2.2.4. Given the significantly shorter randomization time of bacteria in aqueous environments[76], we used a minimum track length threshold of 30 frames. Figure S3, Supporting Information shows how varying the threshold for minimum track length impacts the precision and recall for our method and the baselines. Increasing this threshold will increase the precision while eliminating many bacteria that have shorter tracks due to them being missed in the detection phase, potentially due to very low-frequency motion.

2.3.5. Statistical Analysis

All experimental data were collected from a minimum of two independent experiments. The GT data were produced by an experienced user with less than 2% tracking error. All data were tested for normal distributions. All pairwise comparisons were performed using a *t*-test, and *p*-values smaller than 0.05 were considered significant.

3. Results and Discussion

3.1. Evaluation of Detection Bias

We first inquired if MEMTrack is able to detect and track bacteria from each of the four motility subpopulations equally well. Such performance capability is crucial to the accurate and unbiased representation of population-scale behavior. To evaluate the presence of any biases, we segregated the GT bacteria and the TP tracked bacteria into four subpopulations based on their motility (refer to Section 2.2.4). **Table 1** shows the number of GT and TP bacteria in each of the four categories. The % Detected column shows the fraction of the GT bacteria outputted from MEMTrack. It can be seen that MEMTrack has a comparable detection rate for nearly all categories. The lower detection rate for the medium motility group may be attributed to the small number of bacteria in this subpopulation, which leads to significant fluctuation in the calculation of % Detected.

3.2. MEMTrack's Performance in Collagen

We next conducted ablation studies to evaluate the contribution of each component of the MEMTrack framework to its overall performance (Section S.B, Supporting Information). Table S1, Supporting Information shows the impact of changing the detector model and detection features on the performance of MEMTrack. We see that a single object detector that only uses raw image features underperforms compared to when it incorporates both optical flow (OP) and median background difference (MBD) features, highlighting the efficacy of the motion-enhancer

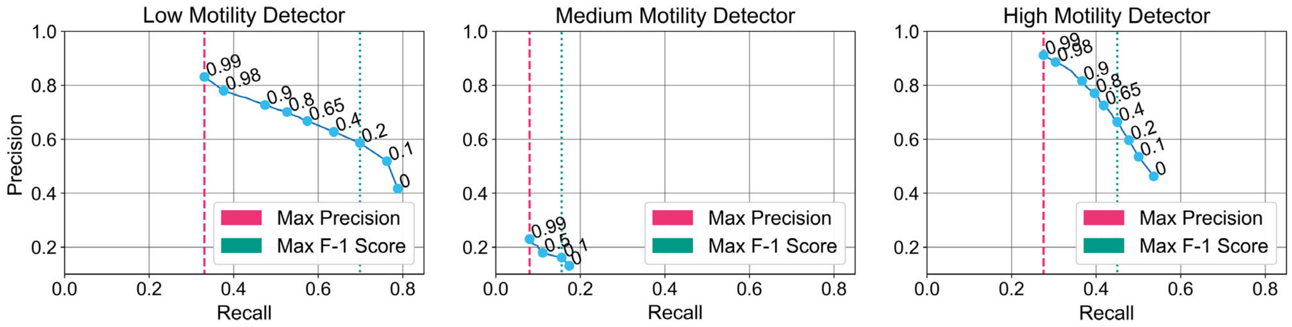

Figure 3. Precision and recall of the three motility detectors (labeled over each plot) for varying confidence score thresholds. Score thresholds of each model based on the maximum precision (pink) and maximum F1 score criteria (blue) are shown as a vertical dashed line and a vertical dotted line, respectively.

Table 1. Fraction of bacteria detected by MEMTrack in each of the four motility subpopulations.

Subpopulation	GT	TP	% Detected
No motility	17	9	53
Low motility	140	66	47
Medium motility	16	5	31
High motility	51	28	55

submodules. Furthermore, while RetinaNet with both OP and MBD features matches MEMTrack in detection performance, it falls short in tracking, mainly due to its single-model approach, which struggles to effectively generate meaningful confidence scores for bacteria of varying motility characteristics, leading to lower precision after tracking.

Table S2, Supporting Information compares the tracking performance of MEMTrack ablations designed to assess the importance of different steps involved in the False Positive Pruner (FPP) and Tracking modules, namely, the use of the Bounding Box (BBox) filter, Confidence Score (Conf.) filter, NMS filter, and Interpolated SORT. Table S2, Supporting Information indicates that omitting steps from the FPP module in MEMTrack significantly reduces precision, underlining the module's crucial role in eliminating FPs. Additionally, the Conf. filter greatly enhances precision, while the BBox and NMS filters have minimal impact. The absence of the interpolated SORT algorithm notably affects recall, demonstrating the essential balance provided by the FPP and Tracking modules in achieving optimal tracking performance. Figure S5, Supporting Information shows the transformation of a representative image through the four modules of MEMTrack.

We next evaluated MEMTrack's performance in detecting and tracking bacteria in collagen (Video S2, Supporting Information). **Figure 4** shows the precision and recall values attained at each step of the pipeline for the test datasets. The precision increases after each False Positive Pruner module filter with a trade-off in the recall. The Interpolated Tracker module further boosts the recall, although at the cost of adding some FPs or a slightly reduced precision. Informed by the physics of bacterial motion in collagen (Section 2.2.4), the Track Length Filter removes tracks shorter than 60 frames and achieves a reasonable balance between precision and recall.

Representative GT and predicted trajectories of bacteria in collagen are shown in **Figure 5**A and Figure S4, Supporting Information. To determine MEMTrack's ability to describe the average speed of the bacterial population accurately, we compared the average speed of the GT bacteria ($8.4 \pm 5.48\,\mu\text{m s}^{-1}$) with the tracked bacteria ($9.7 \pm 7.3\,\mu\text{m s}^{-1}$), as shown in Figure 5B. An unpaired *t*-test yielded a *p*-value of .0792, indicating a lack of significant difference between the two groups; thus, MEMTrack is able to successfully track bacteria and ascertain population-scale motility speed in the dense and low-contrast collagen environment.

We then compared MEMTrack's performance in the collagen environment to four state-of-the-art baseline models, using four criteria of precision, recall, mIOU, and RMSE (**Table 2**). In all cases, predictions from the baseline models were compared against the GT data, followed by filtering out tracklets that have a minimum length of 60 frames. Using the maximum precision criterion, MEMTrack achieved a precision of 76% precision, a recall of 51%, an mIOU of 0.72, and an RMSE of 1.84 μm. Comparatively, the classical computer vision-based methods had lower performance. Trackmate had significantly lower precision and recall of 45% and 17%, respectively. MosaicSuite showed a comparatively high recall at 71% but at the cost of a significantly lower precision of 7% (i.e., only 7% of tracks are TPs), making its results unusable without extensive manual post-processing for filtering out the FPs. It should be noted that MosaicSuite was developed to detect bright spots (i.e., lighter than the background)[18]; therefore, it cannot detect bacteria that appear darker than the background. The background detection-based method, YSMR, also showed much lower precision and recall at 15% and 22%, respectively. The deep learning-based method, DETR (trained on images only), combined with SORT, achieved precision and recall of 79% and 22%, respectively.

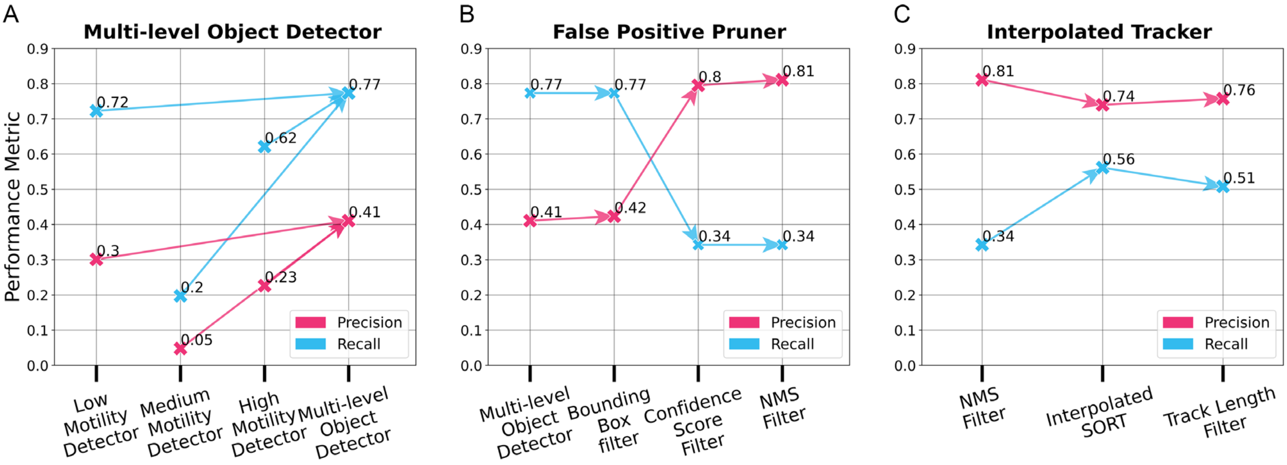

Figure 4. A) Precision and recall for each of the individual motility detectors over their respective GT motility sets and the combined detection result against the entire test set. B) Change in precision and recall at each step of the False Positive Pruner module. C) Precision and recall after tracking and Track Length Filter.

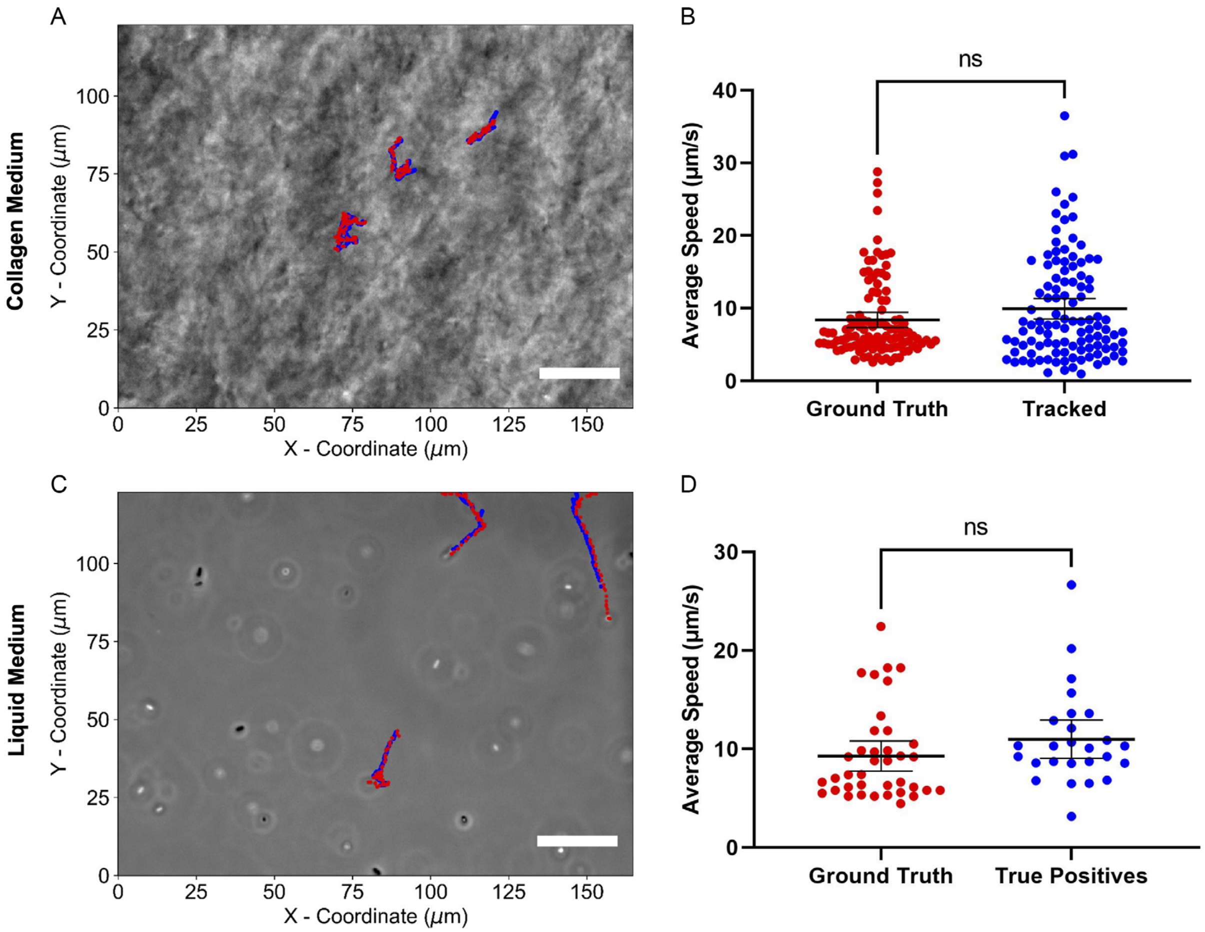

Figure 5. A) Representative GT and corresponding predicted trajectories of bacteria in collagen. B) The average speed of GT bacteria tracked by a human (red) and the trajectories produced by MEMTrack (blue), irrespective of their match to the GT in collagen ($n = 108$ for GT, $n = 108$ for tracked). C) Representative GT and corresponding predicted trajectories of bacteria in liquid medium. D) The average speed of the GT bacteria tracked by a human (red) and the trajectories produced by MEMTrack (blue), irrespective of their match to the GT in liquid medium ($n = 65$ for GT, $n = 38$ for tracked). The scale bars in the microscopy images are 20 μm. ns signifies a lack of statistical significance between the datasets ($p > .05$).

DETR, trained using images and motion features (produced by MEMTrack's Motion Enhancer module), had the highest recall at 86% but its precision was much lower than MEMTrack at 29%. It is important to note that DETR inherently downscales the resolution of image features, a process intended to aid transformer training. This resolution downscaling in DETR might contribute to its suboptimal performance, as it potentially results in the loss of critical details in smaller objects. Overall, our approach provides a minimum of 2.6-fold higher precision than the baseline methods with reasonably high recall. Furthermore, MEMTrack notably outperforms nearly all baseline models, achieving the highest mean mIoU and the lowest RMSE, underscoring its superior accuracy in both bounding box localization and coordinate prediction.

3.3. Applicability of MEMTrack to Aqueous Environments

To assess MEMTrack's generalizability to unseen media, we next determined how MEMTrack, trained on collagen data, performs in commonly used liquid environments. We evaluated the performance of MEMTrack on bacteria motility data collected in an aqueous medium (Video S3, Supporting Information). Representative GT and predicted trajectories of bacteria in liquid are shown in Figure 5C. To determine MEMTrack's ability to describe the average speed of the bacterial population accurately, we compared the average speed of the GT bacteria, $9.2 \pm 4.87\ \mu\text{m s}^{-1}$, with that of the tracked bacteria, $11.0 \pm 5.60\ \mu\text{m s}^{-1}$, as shown in Figure 5D. An unpaired *t*-test comparing the speeds of GT and tracked yielded a *p*-value of .1584, indicating no significant difference between the two groups; thus, MEMTrack can track bacteria successfully and ascertain their population-scale motility speed in the aqueous environment.

Table 3 compares the zero-shot performance of MEMTrack (pretrained in collagen) with the baseline models. Consistent with the results in collagen, MEMTrack performed better than all baseline models with 93% precision and 23% recall using the maximum precision criterion, and 89% precision and 41% recall using the maximum F1 score criterion. Trackmate produced somewhat comparable results with a precision of 83%

Table 2. Comparison of MEMTrack's performance in collagen against baseline models. DETR (Images) was trained only on videos obtained in collagen, while DETR (Images + Motion Features) was trained on features from MEMTrack's Motion Enhancer module.

Method	Tracking		Post Track Length Filter (60)		mIoU	RMSE [μm]
	Precision	Recall	Precision	Recall		
Trackmate	0.43	0.20	0.45	0.17	0.16	5.09
MosaicSuite	0.03	0.94	0.07	0.71	0.28	4.17
YSMR	0.15	0.22	0.15	0.22	0.21	4.70
DETR (Images) + SORT	0.18	0.82	0.22	0.79	0.71	1.93
DETR (Images + Motion Features) + SORT	0.23	0.88	0.29	0.86	0.72	1.77
MEMTrack maximum precision	**0.74**	**0.56**	**0.76**	**0.51**	**0.72**	**1.84**
MEMTrack maximum F1 score	**0.49**	**0.83**	**0.53**	**0.79**	**0.73**	**1.61**

Table 3. Comparison of MEMTrack's zero-shot inference-only performance in liquid media with the baseline models. DETR (Images) was trained only on videos obtained in collagen, while DETR (Images + Motion Features) was trained on features from MEMTrack's Motion Enhancer module.

Method	Tracking		Post Track Length Filter (60)		mIoU	RMSE [μm]
	Precision	Recall	Precision	Recall		
Trackmate	0.84	0.39	0.83	0.37	0.17	4.95
MosaicSuite	0.27	0.94	0.39	0.86	0.22	4.47
YSMR	0.54	0.55	0.54	0.55	0.20	4.67
DETR (Images) + SORT	0.17	0.89	0.20	0.82	0.40	3.62
DETR (Images + Motion Features) + SORT	0.13	0.93	0.19	0.84	0.32	3.62
MEMTrack maximum precision	**0.93**	**0.23**	**0.93**	**0.23**	**0.60**	**1.98**
MEMTrack maximum F1 score	**0.88**	**0.43**	**0.89**	**0.41**	**0.60**	**1.90**

and a recall of 37%. MosaicSuite's precision was 39%, which was significantly improved compared to its performance in collagen (7%); nonetheless, its significantly low precision necessitates manual intervention to remove FPs. YSMR demonstrated moderate performance, achieving 54% precision and 55% recall. In contrast, DETR's (pretrained in collagen) zero-shot performance was weaker, with a 19% precision and 84% recall. Expectedly, applying the Track Length Filter improved the precision in most cases and decreased the recall. Altogether, these results demonstrate that MEMTrack can effectively track micro/nanoscale objects in the previously unseen liquid media without additional training. The performance of MEMTrack and DETR in liquid media can potentially be further improved by retraining models on liquid data instead of using the pretrained Collagen models.

3.4. Effect of Change in Object Size, Shape, and Motion Pattern on MEMTrack Performance

To assess MEMTrack's versatility, we conducted two additional experiments to evaluate MEMTrack's performance in tracking microrobots of varying shapes, sizes, and motion patterns. We chose spherical and prolate spheroid BacteriaBots[77], formed by interfacing several motile bacteria with each microscale object (**Figure 6**A,C). BacteriaBots (characteristic dimension: ≈7.5–10 μm) are significantly larger than bacteria (characteristic dimension: ≈3.5 μm) and have motion characteristics that are distinctly different from those of bacteria.[77–79] As a result, a model trained to detect bacteria is not expected to perform well in detecting BacteriaBots.

To show the adaptability of our MEMTrack framework for BacteriaBots, we implemented a transfer learning approach where the model trained on data from bacteria in collagen was fine-tuned on data from spherical BacteriaBots in the aqueous environment. Transfer learning is a standard procedure in deep learning, where instead of starting the training process from a randomly initialized neural network (i.e., from scratch), pretrained weights learned on a source dataset (in this case, bacteria in collagen data) are used to initialize the neural network. We allocated twelve videos for training and ten videos for testing. All videos were at 20 FPS for 150 s. Given that the spherical BacteriaBots have a narrow speed distribution with values similar to the high motility bacteria subpopulation in collagen, we used a single-model framework for object detection. We fine-tuned the model using the weights or learned parameters from the model trained on high motility bacteria. Following the same methodology (Section 2.1) and maintaining the same hyperparameter thresholds (Section 2.3.4) except the bounding box size which was adjusted for the BacteriaBot size, we generated predictions for the BacteriaBot dataset. We evaluated the results using the same metrics used in evaluating bacteria in collagen and aqueous media. As reported in **Table 4**, we achieved a 99% precision, a 94% recall, an IoU of 0.90, and an RMSE of 0.56 μm for the spherical BacteriaBots using the maximum precision criterion. Additionally, we conducted a zero-shot inference experiment on prolate spheroid BacteriaBots, further testing our fine-tuned model's adaptability without retraining. Our model demonstrated an 82% precision, a 74% recall, a mean IoU of 0.61, and an RMSE of 3.52 μm.

Finally, we compared the average speed of GT and tracked BacteriaBots (Figure 6B,D). For spherical BacteriaBots, the GT average speed of $8.5 \pm 2.1\,\mu\text{m s}^{-1}$ was not significantly different from the predicted speed of $8.7 \pm 3.1\,\mu\text{m s}^{-1}$ ($p = .7777$). Similarly, for prolate spheroid BacteriaBots, the GT average speed of $9.4 \pm 1.51\,\mu\text{m s}^{-1}$was not significantly different from the predicted speed of $9.9 \pm 2.14\,\mu\text{m s}^{-1}$ ($p = .7221$). These results underscore MEMTrack's versatility for diverse tracking scenarios.

4. Limitations and Future Work

MEMTrack, while effective for various object shapes and sizes and in various media, has limitations that present avenues for future research. A primary limitation is its reduced efficacy in tracking low-motility microrobots. This is due to MEMTrack's

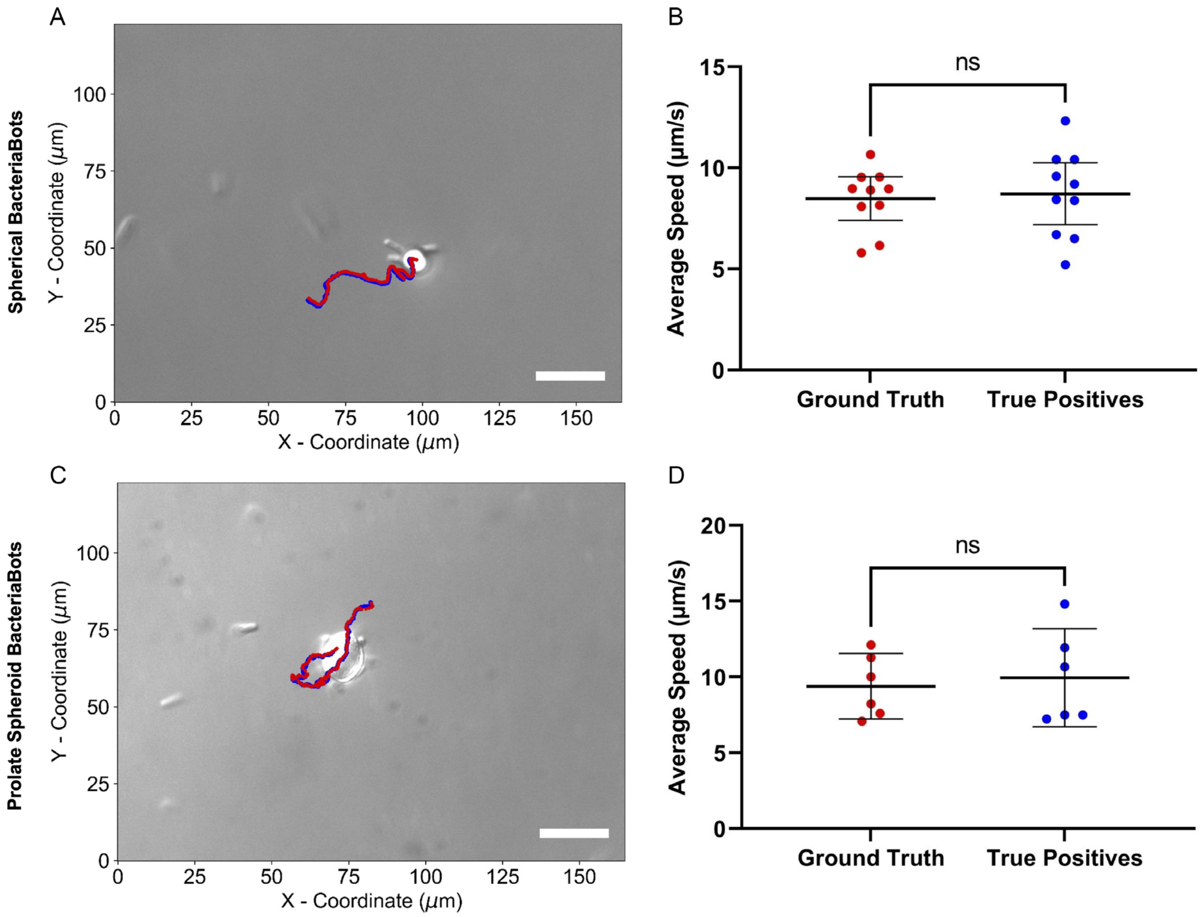

Figure 6. A) Representative GT and corresponding predicted trajectories of spherical BacteriaBots in liquid medium. B) The average speed of the GT spherical BacteriaBots tracked by a human (red) and the trajectories produced by MEMTrack (blue), irrespective of their match to the GT ($n = 10$ for GT, $n = 10$ for tracked). C) Representative GT and corresponding predicted trajectories of prolate spheroid BacteriaBots in liquid medium. D) The average speed of the GT prolate spheroid BacteriaBots tracked by a human (red) and the trajectories produced by MEMTrack (blue), irrespective of their match to the GT for prolate spheroid BacteriaBots in liquid medium ($n = 6$ for GT, $n = 6$ for tracked). The scale bars in the microscopy images are 20 μm. ns signifies a lack of statistical significance between the datasets ($p > .05$).

Table 4. MEMTrack's performance on spherical (inference from the trained model) and prolate spheroid (zero-shot analysis) BacteriaBots.

Test data	Test mode	Criteria	Precision	Recall	mIoU	RMSE [μm]
Spherical BacteriaBots	Trained model	Maximum precision	0.99	0.94	0.90	0.56
Spherical BacteriaBots	Trained model	Maximum F1	0.99	0.96	0.90	0.56
Prolate spheroid BacteriaBots	Zeroshot	Maximum precision	0.82	0.74	0.61	3.52
Prolate spheroid BacteriaBots	Zeroshot	Maximum F1	0.82	0.77	0.62	3.47

core design focusing on motion enhancement, which makes it less suited for detecting and tracking objects exhibiting minimal or no movement. Advancing our method to detect and accurately track low-frequency motion patterns is a key area for improvement. Additionally, MEMTrack currently employs the SORT algorithm for tracking and handling missed detections, which operates under the assumption that the state variables of the objects (e.g., velocities) change linearly over time. However, as self-propelled microrobots often exhibit random movement, incorporating a physics-based model in the SORT algorithm can improve MEMTrack's performance by better accommodating the stochastic motion of microrobots. Finally, although MEMTrack shows superior precision and recall metrics compared to baseline methods, there is significant room for improvement in the challenging problem of detecting and tracking microrobots in dense, low-contrast environments. Future work can focus on developing detection modules that use temporal information in long sequences of frames with the help of transformer networks to detect microrobots.

5. Conclusion

In this work, we present MEMTrack, an automated pipeline for detecting and tracking microrobots in dense and low-contrast environments, such as collagen. This is a particularly challenging problem given the lack of visual features distinguishing the foreground objects from the background. Our approach leverages synthetic motion features, the RetinaNet object detection model, and a modified SORT tracking algorithm with interpolation to achieve robust results against different background media at high precision. Our results demonstrate that the proposed pipeline is versatile and adaptable to different tracking scenarios. It can robustly substitute the tedious task of manually tracking microrobots for predicting population-scale speed values in different backgrounds (i.e., collagen and liquid environments) and for different size and shape microrobots.

We envision MEMTrack to impact both fundamental and translational microrobotic research. In fundamental research, high spatiotemporal resolution imaging is used to develop an understanding of the physics of propulsion, swarm dynamics (e.g., agent–agent interaction), and experimental evaluation of centralized and decentralized control strategies. The resulting images, often collected using bright-field microscopy, can be low-contrast and difficult to process automatically. Bright-field images collected in tissue phantoms (e.g., collagen, Figure 1) or non-Newtonian biofluids (e.g., mucus) are even more challenging to process automatically. Thus, robustly performing multiobject trackers such as MEMTrack could have an immediate impact on accelerating discoveries and advancing the fundamental knowledge in the field. Microrobotic technology translational efforts are currently centered around biomedical applications. For in vivo applications, microrobots are imaged using MRI[80] or ultrasound,[81] which also produce grayscale and sometimes low-contrast images. As a result, MEMTrack could also find applications in investigating the biodistribution of microrobots in complex in vivo environments.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

The authors acknowledge Behkam Lab member Ying Zhan for sharing previously published bacteria aqueous swimming data to test MEMTrack performance in speed predictions. This research was partly supported by NSF grants CBET-2133739 and CBET-1454226 to BB, 4-VA grant to B.B., and NSF grant IIS-2107332 to A.K. The Advanced Research Computing (ARC) Center at Virginia Tech provided access to computing resources. [Correction added on 21 March 2024 after online publication: Typos in text were updated in this version.]

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

M.S. and B.K. contributed equally to this work. B.B. conceived and supervised the research. A.K. supervised the machine learning component of the research. M.S., B.K., A.K., and B.B. designed the research methodology. M.S., A.D., and A.K. developed the MEMTrack pipeline. B.K. and E.J.L. conducted the experiments in collagen and aqueous media. B.K. generated the training, testing, and validation datasets. All authors contributed to the data analysis. M.S., B.K., A.K., and B.B. wrote the manuscript. All authors edited the manuscript.

Data Availability Statement

All source codes for training and testing MEMTrack and reproducing the results reported in this manuscript are publicly available at https://github.com/KGML-lab/MEMTrack.

Keywords

bacteria, biohybrid microrobotics, collagen, computer vision, machine learning, multiobject tracking, object detection

Received: September 20, 2023
Revised: December 30, 2023
Published online:

[1] B. Wang, K. Kostarelos, B. J. J. Nelson, L. Zhang, *Adv. Mater.* **2021**, *33*, 2002047.
[2] D. Li, C. Liu, Y. Yang, L. Wang, Y. Shen, *Light Sci. Appl.* **2020**, *9*, 84.
[3] J. Jiang, Z. Yang, A. Ferreira, L. Zhang, *Adv. Intell. Syst.* **2022**, *4*, 2100279.
[4] F. Soto, J. Wang, R. Ahmed, U. Demirci, *Adv. Sci.* **2020**, *7*, 2002203.
[5] A. Aziz, M. Medina-Sánchez, J. Claussen, O. G. G. Schmidt, *Nano Lett.* **2019**, *19*, 6612.
[6] K. Botros, M. Alkhatib, D. Folio, A. Ferreira, *IEEE Robot. Autom. Lett.* **2023**, *8*, 1.
[7] M. E. E. Tiryaki, S. O. O. Demir, M. Sitti, *IEEE Robot. Autom. Lett.* **2022**, *7*, 6982.
[8] S. Pane, V. Iacovacci, E. Sinibaldi, A. Menciassi, *Appl. Phys. Lett.* **2021**, *118*, 14102.
[9] A. Aziz, S. Pane, V. Iacovacci, N. Koukourakis, J. Czarske, A. Menciassi, M. Medina-Sánchez, O. G. G. Schmidt, *ACS Nano* **2020**, *14*, 10865.
[10] J. P. P. Bae, S. Yoon, M. Vania, D. Lee, *Int. J. Control. Autom. Syst.* **2020**, *18*, 21.
[11] Q. Wang, L. Zhang, *IEEE Open J. Nanotechnol.* **2020**, *1*, 6.
[12] C. A. A. Schneider, W. S. S. Rasband, K. W. W. Eliceiri, *Nat. Methods* **2012**, *9*, 671.
[13] J. Y. Y. Tinevez, N. Perry, J. Schindelin, G. M. M. Hoopes, G. D. D. Reynolds, E. Laplantine, S. Y. Y. Bednarek, S. L. L. Shorte, K. W. W. Eliceiri, *Methods* **2017**, *115*, 80.
[14] D. Ershov, M. S. S. Phan, J. W. W. Pylvänäinen, S. U. U. Rigaud, L. Le Blanc, A. Charles-Orszag, J. R. W. R. W. Conway, R. F. F. Laine, N. H. H. Roy, D. Bonazzi, G. Duménil, G. Jacquemet, J. Y. Y. Tinevez, *Nat. Methods* **2022**, *19*, 829.
[15] E. Meijering, O. Dzyubachyk, I. Smal, *Methods Enzymol.* **2012**, *504*, 183.
[16] C. McQuin, A. Goodman, V. Chernyshev, L. Kamentsky, B. A. A. Cimini, K. W. W. Karhohs, M. Doan, L. Ding, S. M. M. Rafelski, D. Thirstrup, W. Wiegraebe, S. Singh, T. Becker, J. C. C. Caicedo, A. E. E. Carpenter, *PLoS Biol.* **2018**, *16*, 2005970.

[17] D. R. R. Stirling, M. J. J. Swain-Bowden, A. M. M. Lucas, A. E. E. Carpenter, B. A. A. Cimini, A. Goodman, *BMC Bioinform.* **2021**, *22*, 433.
[18] I. F. F. Sbalzarini, P. Koumoutsakos, *J. Struct. Biol.* **2005**, *151*, 182.
[19] C. S. S. Bjornsson, G. Lin, Y. Al-Kofahi, A. Narayanaswamy, K. L. L. Smith, W. Shain, B. Roysam, *J. Neurosci. Methods* **2008**, *170*, 165.
[20] P. Kankaanpää, L. Paavolainen, S. Tiitta, M. Karjalainen, J. Päivärinne, J. Nieminen, V. Marjomäki, J. Heino, D. J. J. White, *Nat. Methods* **2012**, *9*, 683.
[21] F. De Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, *Nat. Methods* **2012**, *9*, 690.
[22] A. D. D. Balomenos, P. Tsakanikas, Z. Aspridou, A. P. P. Tampakaki, K. P. P. Koutsoumanis, E. S. S. Manolakos, *BMC Syst. Biol.* **2017**, *11*, 43.
[23] Q. Wang, J. Niemi, C.-M. Tan, L. You, M. West, *Cytom. Part A* **2010**, *77*, 101.
[24] J. Klein, S. Leupold, I. Biegler, R. Biedendieck, R. Münch, J. Dieter, *Bioinformatics* **2012**, *28*, 2276.
[25] J. A. A. Cornwell, J. Li, S. Mahadevan, J. S. S. Draper, G. L. L. Joun, H. Zoellner, N. S. S. Asli, R. P. P. Harvey, R. E. E. Nordon, *SoftwareX* **2020**, *11*, 100440.
[26] M. J. J. Rust, M. Lakadamyali, B. Brandenburg, X. Zhuang, *Cold Spring Harb. Protoc.* **2011**, *2011*, 1978.
[27] L. Feng, Y. Xu, Y. Yang, X. Zheng, *J. Struct. Biol.* **2011**, *173*, 219.
[28] H. M. M. Van Der Schaar, M. J. J. Rust, C. Chen, H. van der Ende-Metselaar, J. Wilschut, X. Zhuang, J. M. M. Smit, *PLoS Pathog.* **2008**, *4*, 1000244.
[29] S. Stylianidou, C. Brennan, S. B. B. Nissen, N. J. J. Kuwada, P. A. A. Wiggins, *Mol. Microbiol.* **2016**, *102*, 690.
[30] L. Gardini, M. Capitanio, F. S. S. Pavone, *Sci. Rep.* **2015**, *5*, 1.
[31] D. Yuan, X. Chang, Q. Liu, Y. Yang, D. Wang, M. Shu, Z. He, G. Shi, *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, May **2023**, pp. 1-13. https://ieeexplore.ieee.org/abstract/document/10122634.
[32] D. Yuan, X. Chang, Z. Li, Z. He, *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**, *18*, 1.
[33] D. Yuan, X. Chang, P. Y. Huang, Q. Liu, Z. He, *IEEE Trans. Image Process.* **2021**, *30*, 976.
[34] P. Vallotton, A. M. M. Van Oijen, C. B. B. Whitchurch, V. Gelfand, L. Yeo, G. Tsiavaliaris, S. Heinrich, E. Dultz, K. Weis, D. Grünwald, *Traffic* **2017**, *18*, 840.
[35] C. Spahn, E. Gómez-de-Mariscal, R. F. F. Laine, P. M. M. Pereira, L. von Chamier, M. Conduit, M. G. G. Pinho, G. Jacquemet, S. Holden, M. Heilemann, *Commun. Biol.* **2022**, *5*, 688.
[36] J. Schwanbeck, I. Oehmig, J. Dretzke, A. E. E. Zautner, U. Groß, W. Bohne, *BMC Bioinform.* **2020**, *21*, 1.
[37] J. Xie, S. Khan, M. Shah, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008: 11th Int. Conf.*, Springer, New York, NY **2008**, pp. 824–832.
[38] S. B. B. Mehta, M. McQuilken, P. J. J. La Riviere, P. Occhipinti, A. Verma, R. Oldenbourg, A. S. S. Gladfelter, T. Tani, *Proc. Natl. Acad. Sci.* **2016**, *113*, 6352.
[39] J. M. M. Newby, A. M. M. Schaefer, P. T. T. Lee, M. G. G. Forest, S. K. K. Lai, *Proc. Natl. Acad. Sci.* **2018**, *115*, 9026.
[40] D. E. E. Hernandez, S. W. W. Chen, E. E. E. Hunter, E. B. B. Steager, V. Kumar, presented at *MARSS 2018—Int. Conf. on Manipulation, Automation and Robotics at Small Scales*, Nagoya, July 2018.
[41] E. Pottash, R. McKay, C. Virgile, H. Ueda, W. Bentley, *Biotechniques* **2017**, *62*, 31.
[42] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler (Preprint), arXiv:1603.00831, v1, submitted: Mar. **2016**.
[43] V. A. A. Webster-Wood, M. Guix, N. W. W. Xu, B. Behkam, H. Sato, D. Sarkar, S. Sanchez, M. Shimizu, K. K. K. Parker, *Bioinspir. Biomim.* **2023**, *18*, 15001.
[44] B. K. P. K. P. Horn, B. G. G. Schunck, *Artif. Intell.* **1981**, *17*, 185.
[45] B. D. D. Lucas, T. Kanade, in *IJCAI'81: 7th Int. Joint Conf. on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA **1981**, pp. 674–679
[46] R. Girshick, J. Donahue, T. Darrell, J. Malik, *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142.
[47] R. Girshick, in *Proc. of the IEEE Int. Conf. on Computer Vision* **2015**, IEEE pp. 1440–1448,.
[48] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE **2016**, pp. 779–788.
[49] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. M. Liao (Preprint), arXiv:2004.10934, v1, submitted: Apr. **2020**.
[50] J. Redmon, A. Farhadi, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE **2017**, pp. 7263–7271.
[51] C. Y. Y. Wang, A. Bochkovskiy, H. Y. M. Y. M. Liao (Preprint), arXiv: 2207.02696, v1, submitted: Jul. **2022**.
[52] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, J. Han, *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13467.
[53] F. O. Unel, B. O. Ozkalayci, C. Cigla, in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, IEEE Computer Society **2019**, pp. 582–591.
[54] Z. Q. Zhao, P. Zheng, S. T. Xu, X. Wu, *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *30*, 3212.
[55] M. Muzammul, X. Li, arXiv **2021**.
[56] C. Yang, Z. Huang, N. Wang, in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society **2022**, pp. 13658–13667.
[57] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, P. Dollar, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE **2020**, pp. 318–327
[58] D. S. S. Bolme, J. R. R. Beveridge, B. A. A. Draper, Y. M. M. Lui, in *2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* **2010**, pp. 2544–2550, IEEE.
[59] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, in *2016 IEEE Int. Conf. on Image Processing (ICIP)*, IEEE **2016**, pp. 3464–3468.
[60] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, P. H. S. H. S. Torr, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* **2017**, pp. 2805–2813
[61] N. Wojke, A. Bewley, D. Paulus, in *2017 IEEE Int. Conf. on Image Processing (ICIP)*, IEEE **2017**, pp. 3645–3649.
[62] R. E. E. Kalman, *J. Basic Eng.* **1961**, *83*, 95.
[63] A. Frank, *Nav. Res. Logist.* **2005**, *52*, 2.
[64] H. W. W. Kuhn, *Nav. Res. Logist. Q.* **1955**, *2*, 83.
[65] Y. Zhan, A. Fergusson, L. R. McNally, R. M. Davis, B. Behkam, *Adv. Intell. Syst.* **2022**, *4*, 2100135.
[66] M. A. Traore, C. M. Damico, B. Behkam, *Appl. Phys. Lett.* **2014**, *105*, 173702.
[67] K. M. Broadway, S. Suh, B. Behkam, B. E. Scharf, *J. Biotechnol.* **2017**, *251*, 76.
[68] H. C. Berg, in *Random Walks in Biology*, Princeton University Press, Princeton, New Jersey **1984**.
[69] P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, in *Introduction To Data Mining*, Pearson **2018**, https://www.pearson.com/en-us/subject-catalog/p/introduction-to-data-mining/P200000003204/9780133128901?creative=&keyword=&matchtype=&network=x&device=c&gad_source=1&gclid=CjwKCAiAq4KuBhA6EiwArMAw1OoZIFkCNMQhGt1J9uNcJbVaXJY-akt6z9MqSoyOJ89wYKG2tF1xGBoCCWcQAvD_BwE&gclsrc=aw.ds.
[70] D. Sage, F. R. Neumann, F. Hediger, S. M. Gasser, M. Unser, *IEEE Trans. Image Process.* **2005**, *14*, 1372.

[71] D. Marr, E. Hildreth, *Proc. R. Soc. London. Ser. B. Biol. Sci.* **1980**, *207*, 187.

[72] R. E. Burkard, E. Cela, in *Handbook of Combinational Optimization*, Springer, Boston, MA **1999**, pp. 75–149.

[73] R. Jonker, A. Volgenant, *Computing* **1987**, *38*, 325.

[74] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, in *European Conf. on Computer Vision—ECCV 2020*, Springer, Cham **2020**, pp. 213–229.

[75] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, In *Conf. on Neural Information Processing Systems*, Long Beach, CA, December **2017**.

[76] E. J. Leaman, A. Sahari, M. A. Traore, B. Q. Geuther, C. M. Morrow, B. Behkam, *APL Bioeng.* **2020**, *4*, 016104.

[77] B. Sahari, A. Headen, D. Behkam, A. Sahari, D. Headen, B. Behkam, *Biomed. Microdevices* **2012**, *14*, 999.

[78] M. A. Traoré, A. Sahari, B. Behkam, *Phys. Rev. E* **2011**, *84*, 061908.

[79] A. Sahari, M. A. Traore, B. E. Scharf, B. Behkam, *Biomed. Microdevices* **2014**, *16*, 717.

[80] J.-B. Mathieu, G. Beaudoin, S. Martel, *IEEE Trans. Biomed. Eng.* **2006**, *53*, 292.

[81] S. Mohanty, I. S. M. Khalil, S. Misra, *Proc. R. Soc. A.* **2020**, *476*, 20200621.