Regular Article

Check for
updates

Tracking droplets in soft granular flows with deep learning techniques

Mihir Durve[1,a] , **Fabio Bonaccorso**[1,2,3], **Andrea Montessori**[2] ,
Marco Lauricella[2] , **Adriano Tiribocchi**[2] , **Sauro Succi**[1,2,4]

[1] Center for Life Nano- & Neuro-Science, Fondazione Istituto Italiano di Tecnologia (IIT), 00161 Rome, Italy
[2] Istituto per le Applicazioni del Calcolo CNR, via dei Taurini 19, Rome, Italy
[3] Department of Physics and INFN, University of Rome Tor Vergata, Via della Ricerca Scientifica 1, 00133 Rome, Italy
[4] Institute for Applied Computational Science, John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, USA

© The Author(s) 2021

Abstract The state-of-the-art deep learning-based object recognition YOLO algorithm and object tracking DeepSORT algorithm are combined to analyze digital images from fluid dynamic simulations of multi-core emulsions and soft flowing crystals and to track moving droplets within these complex flows. The YOLO network was trained to recognize the droplets with synthetically prepared data, thereby bypassing the labor-intensive data acquisition process. In both applications, the trained YOLO + DeepSORT procedure performs with high accuracy on the real data from the fluid simulations, with low error levels in the inferred trajectories of the droplets and independently computed ground truth. Moreover, using commonly used desktop GPUs, the developed application is capable of analyzing data at speeds that exceed the typical image acquisition rates of digital cameras (30 fps), opening the interesting prospect of realizing a low-cost and practical tool to study systems with many moving objects, mostly but not exclusively, biological ones. Besides its practical applications, the procedure presented here marks the first step towards the automatic extraction of effective equations of motion of many-body soft flowing systems.

1 Introduction

Over the last decade, machine learning has taken the scientific world by storm and even more so the commercial one. Even though it is not all old gold that glitters [1], the potential of machine learning to gain new insights into a variety of complex systems in science and society is unquestionably tantalizing [2,3].

Deep learning is a sub-field of machine learning specifically inspired by the structure and function of the human brain [4]. Its core consists of artificial neural networks that imitate the operation of the human brain in data processing and decision making. In the actual digital era, which generates an unprecedented amount of data (Big Data) in the form of images, text, and audio-video clips, deep learning techniques have proved capable of extracting relevant information and correlations [4], sometimes dramatically abating the amount of time required

[a] e-mail: mihirdurve@gmail.com (corresponding author)

Springer

by humans or classical algorithms to perform a similar task. This automation in learning from the data has led to the development of many useful computer applications, such as handwriting reading [5,6], human speech analysis [7,8], text sentiment analysis of posts on social media platforms [9–11], to name a few. In recent years, deep learning networks are being also used to study biological and physical systems [12,13]. In microfluidics, for example, deep learning networks were employed to learn physical parameters, study the size and shapes of the droplets in emulsions [14,15].

Nowadays, digital cameras assisted by deep learning are used everywhere for traffic management, surveillance, crowd management, automated billing, and customer services [16–19]. Given a video feed from these cameras, object detection and subsequent tracking of the detected objects are the two most basic tasks in computer vision before carrying out any further analysis. To achieve these tasks, deep learning algorithms are the fundamental tools.

In recent years, multi-layer convolutional neural networks (CNN) have a surge of popularity for feature extraction to accomplish object identification, classification, and tracking [20–23]. The output of the object detection network typically consists of bounding boxes encapsulating the identified objects for visual inspection. The tracking network across sequential images then processes the detected objects to construct tracks of moving objects. State-of-the-art deep learning models for object detection have demonstrated remarkable accuracy compared with classical algorithms and human performance [23,24].

For the first task of object detection, various deep learning-based models differ in their network architecture, such as the number of convolutional layers, number of pooling layers, filter sizes, and the base algorithm used to analyze the input. At present, Faster region-based convolutional neural networks (FRCNNs) [21], You Only Look Once (YOLO) [23], Single Shot Detector (SSD) [25] are state of the art in the field. Out of these models, we adapt the YOLO algorithm for object detection due to its ease of training and superior image analysis speed, which is further scalable using GPUs. For the second task of tracking the detected objects, we employ a state-of-the-art DeepSORT algorithm [22], which uses the appearance of the objects to track them across a sequence of frames. Recently, YOLO and DeepSORT algorithms have been deployed for several real-world applications, such as monitoring covid-19 protocols in real-time [26,27], tracking ball or player trajectories in sports [28,29], and many others.

In this work, we develop YOLO + DeepSORT based droplet recognition application to analyze the multi-core emulsions and soft granular media simulated via Lattice Boltzmann (LB) methods. We train the object detection network to identify droplets from digital images with a synthetically prepared dataset, thereby avoiding the labor-intensive training data gathering process. In principle, the developed application can also be used to analyze the data generated by the experimental setup of similar physical systems. The goal is to extract trajectories of the center of mass of individual droplets as they move within the flow and then use this information to analyze the dynamics of this complex many-body system. The application developed in this work can extract trajectories of the droplets in real-time by analyzing the video of the system.

The paper is organized as follows. Before we dive into details of the YOLO + DeepSORT based droplet recognition application, in the next section, we provide a brief account of the Lattice Boltzmann method and relevant simulation details. In Sect. 3, we describe the YOLO and the DeepSORT methods and how the training is implemented. In Sect. 4, we report our results on a four-core emulsion and a more complex soft granular material, focusing on the accuracy of the YOLO + DeepSORT procedure in trajectory extraction. Finally, we analyze extracted trajectories of the droplets in a dense emulsion system and compare them with apparently similar active matter systems such as a flock of birds.
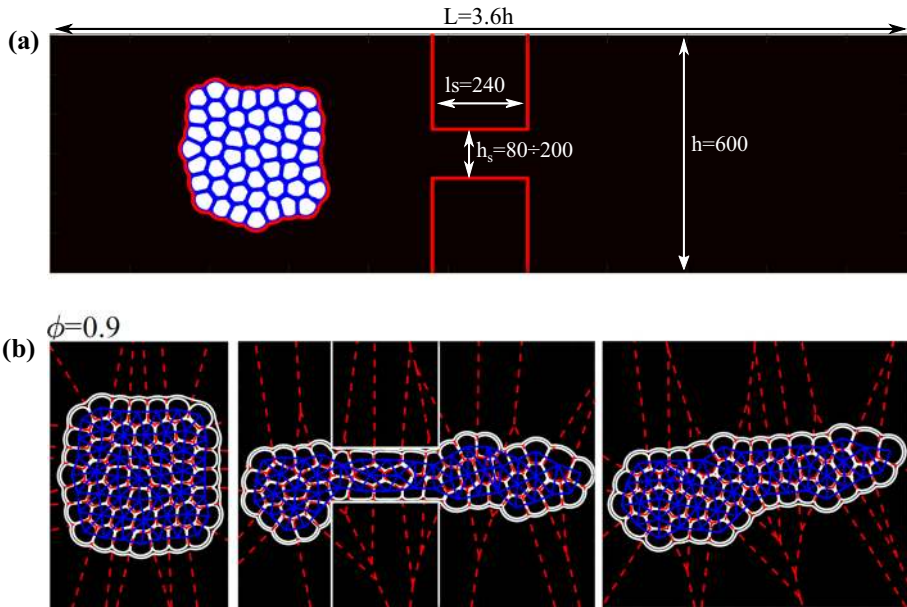
Fig. 1 **a** A triple emulsion with $\phi \simeq 0.9$ and $N_d = 49$ initialized within the inlet chamber. **b** Translocation dynamics of a dense emulsion $\phi = 0.9$, $N_d = 49$ and $h_s/D \simeq 0.45$. Once the emulsion crosses the constriction, the resulting shape observed in the outlet chamber crucially depends on the area fraction ϕ

2 Lattice Boltzmann method and simulations

We employed two variants of the Lattice Boltzmann approach for the multicomponent flows simulations, namely the color-gradient approach with near-contact interactions and the free energy model for multicomponent fluids. In Appendices A and B, we briefly outline the two models, while in the following sections, we report the simulation details for the two physical systems under consideration.

2.1 Simulation details for translocation of a soft granular material within narrow channels

This system is studied using the color gradient LB. The simulation setup consists of a two-dimensional microfluidic channel made of an inlet reservoir followed by a thinner channel connected to a further downstream reservoir. The height of the chambers is $h = 600$ lattice units and the one of the constriction is $h_s = 120$ lattice units while its length is $l_s = 240$ lattice units. We impose a bounce-back rule for the distribution functions at the walls, while at the outlet, we employ absorbing (zero gradient) [30].

The soft granular material is formed by droplets (component A, white region in Fig. 1) immersed within an inter-droplet continuous phase (component B, blue lines) surrounded by an external bulk phase (component C, black region outside the emulsion). Its structure closely resembles that of a high internal phase double emulsion with multi-core morphology [31,32]. In Fig. 1a we show a snapshot of the domain with the relevant dimension of the geometrical setup and in Fig. 1b a translocation sequence of the soft material made of $N_d = 49$ internal droplets occupying an area fraction $\phi \simeq 0.9$.

In our numerical experiment, a uniform velocity profile of speed $U_{in} = 2 \times 10^{-3}$ (in simulation units) is imposed at the inlet, which pushes the emulsion within the narrow constriction. Finally, the value of the velocity ensures that Capillary and Reynolds numbers remain well within the typical range of microfluidic experiments ($Ca \sim \mathcal{O}(10^{-3})$ and $Re \sim \mathcal{O}(1)$).

2.2 Simulation details for a flow of a multi-core emulsions within a microchannel.

Here we report the numerical details of the multi-core emulsion simulated using the free-energy LB. The droplet is set between two flat walls placed at distance L_y, where no-slip conditions hold for the velocity field **v** and neutral wetting for the fluid density ψ. The former means that the velocity is zero at the boundaries, while the latter that no mass flux crosses the walls where droplet interfaces are perpendicular.

In Fig. 2 we show two examples of multi-core emulsions whose design is inspired by experimental realizations. They are made of monodisperse immiscible droplets arranged in a symmetric configuration. In Fig. 2a, three cores (white) of diameter $D_i = 30$ lattice sites are accommodated within a larger drop (black) of diameter $D_O = 100$ lattice sites, in turn, surrounded by a further fluid (white). An analogous setup is used for the four-core emulsion (see Fig. 2b). Starting from these configurations, the emulsions are then driven out of equilibrium by applying a symmetric shear flow, where the top wall moves rightwards (along the positive x axis) at constant speed v_w and the bottom wall in the opposite direction at speed $-v_w$. This sets a shear rate of $\dot{\gamma} = 2L_y/v_w$. In previous works [33,34], it has been shown that, once the shear is turned on, the external shell elongates and aligns along the direction imposed by the flow, while the internal cores acquire a periodic rotational motion triggered by a fluid vortex formed within the emulsion. These dynamics persist over long periods of time provided that the shear is on. In Fig. 2c, d we show two instantaneous configurations of a three and a four-core emulsion at late times, where $\dot{\gamma} \simeq 10^{-4}$ with $v_w = 0.01$ and $L_y = 170$ in simulation units.

This system approximately corresponds to a real multi-core emulsion with a shell of diameter $\simeq 100\mu$m and cores of diameter $\simeq 30\mu$m, having a surface tension ranging between $1 - 10$mN/m, a viscosity $\simeq 10^{-1}$ Pa·s and a shear rates varying between $0.1 - 1/$s. Like in the previous method, Capillary and Reynolds numbers here range between 0.1 and 1 as well, as in typical microfluidic experiments.

The output of the simulations of the physical systems described above are saved as video files and later given as input to the deep learning-based application for automatic extraction of the trajectories of the moving droplets. In the supplementary material [35], see video1.avi file for the simulation output of translocation of triple emulsions within narrow channels and video2.avi for a flow of multi-core emulsions within a microchannel.

3 Technique

In this work, we develop an application to extract trajectories of individual droplets simulated via LB methods by adapting two different algorithms, one for object detection and another one for object tracking. The first task consists of training an artificial neural network to analyze digital images and locate the objects of interest, which in this case are the droplets. Later, the second task is to track the located objects across multiple sequential frames to infer their trajectories. Figure 3 shows the typical steps involved to accomplish the object recognition task [36]. In the following section, we briefly describe the algorithm employed for the object recognition.
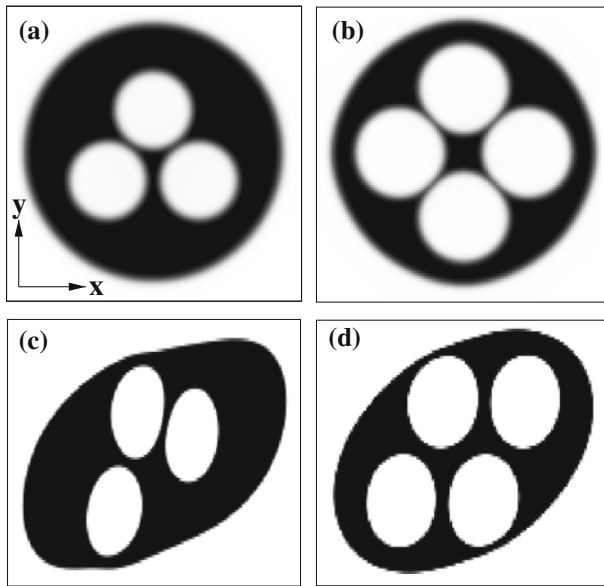
Fig. 2 **a**, **b** Initial configurations of a three and a four-core monodisperse emulsion. The diameter of the cores is $D_i = 30$ lattice sites while that of the shell is $D_O = 100$ lattice sites. **c**, **d** Examples of instantaneous configurations of a the multi-core emulsions subject to a symmetric shear flow. The shear rate is $\dot{\gamma} \simeq 10^{-4}$ in simulation units. In both cases internal cores rotate periodically clockwise, following an almost elliptical path triggered by the fluid vortex confined within the shell. For more details see Ref.[33,34]. Here $Ca \simeq 0.2$ and $Re \simeq 1.2$
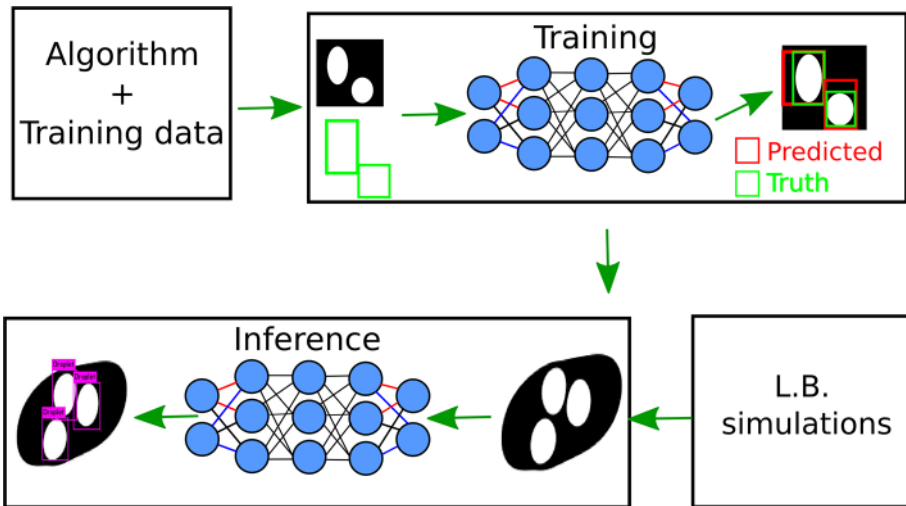

Fig. 3 Typical steps in developing the deep learning-based application for object detection. In the training phase, difference between prediction and truth is used to improve the accuracy of the network by updating the parameters of the network. At the inference stage, a real world data, in our case from LB simulation, is fed to the network to obtain the bounding boxes around the droplets
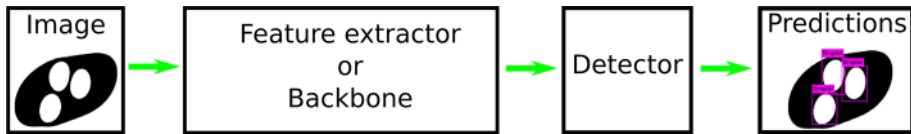
Fig. 4 High level sketch of the YOLO network structure

3.1 You only look once (YOLO)

You Only Look Once (YOLO) is a state-of-the-art algorithm that employs a single multi-layer network for object identification and classification. The YOLO algorithm has shown remarkable accuracy with high processing speed in identifying and classifying objects from the COCO dataset, which consists of 80 types (classes) of everyday objects [23].

The building structure of the YOLO network consists of two main components, a feature extractor and a detector. When an image is given as an input to the YOLO network, the extractor (also called backbone) produces feature representations at different scales (see Fig. 4). Such representations are then passed to the detector that outputs the bounding boxes along with the confidence score and the classes of the detected objects. In this work, we use the YOLO algorithm to identify droplets in the digital images produced by LB simulations of two physical systems. We employ the YOLO-v3 network, which uses the Darknet-53 network as a backbone. The Darknet-53 is a deep network consisting of 53 layers, which has shown significant improvement in object detection over its predecessors YOLO-v2 and YOLO-v1, which employ backbone with fewer layers [24].

The YOLO algorithm achieves high speed of image analysis due to its operating principle, namely by treating object detection as a regression instead of a classification task in determining class probabilities to the bounding boxes. The operating procedure of the YOLOv3 algorithm is sketched in Fig. 5. All objects in the given images are located and classified simultaneously. To achieve that, the YOLO algorithm divides the input image into $S \times S$ grid cells. Each grid cell is responsible for detecting an object if the object's center is within the cell domain. Each grid cell predicts B bounding boxes with their confidence score for the object and C conditional class probabilities for the given object belonging to a specific class. This information is then combined to produce the final output as a single bounding box around the detected object along with the class of that object. The final output is then passed to the DeepSORT algorithm for tracking the droplets.

3.2 DeepSORT

DeepSORT is an algorithm that tracks detected objects between two successive frames [22]. By analyzing sequential frames, DeepSORT can construct trajectories of individual objects. It uses the Hungarian algorithm to distinguish the objects detected in two consecutive frames and assigns individual objects their unique identity. Kalman filtering is then used to predict the future position of the objects based on their current positions.

The YOLO and the DeepSORT algorithms together accomplish droplet recognition and tracking from the simulated systems. In the next section, we outline the training data acquisition process employed in this work and later describe the training process for the YOLO network for droplet detection.
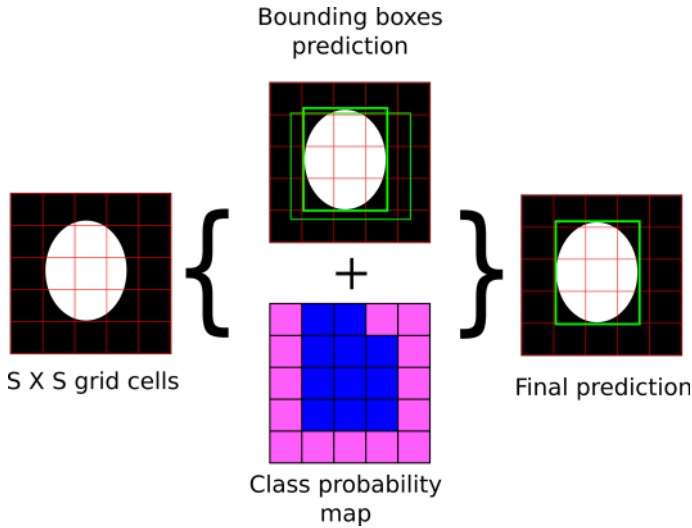
Bounding boxes prediction

S X S grid cells

+

Class probability map

Final prediction

Fig. 5 Operating principle of the YOLO algorithm. Initially, an input image is split into $S \times S$ grid. In the next stage, multiple bounding boxes are predicted along with a class probability map (blue color highlights cells hosting an object belonging to the "droplet" class) to give final bounding boxes and classes of objects

3.3 Training data

Before training a network, acquiring training data is a crucial part of an object detection network. Training data consists of several images of the objects to be detected along with information about their location and dimensions. Practically, acquiring the training data is a labor-intensive task involving gathering the images and manually marking the positions and dimensions of the objects. For example, gathering thousands of images of street view cameras and marking the location of the cars to train a network to identify cars in a video feed. In addition, the prospect of manual marking of the objects makes the training data susceptible to human errors.

We avoided the labor-intensive process by preparing a synthetic training data set. In this dataset, each image is a collage with few solid white circles, mimicking the droplets placed randomly on a dark uniform background. The solid white circles of different sizes and shapes are prepared with commonly used computer graphics software. A separate text file is generated to note the positions and dimensions of the randomly placed droplets. A Python script generates several thousand images and associated text files in a matter of few seconds. The script is provided in the supplementary material [35]. We prepared several types of training datasets, and the detailed description is given in Appendix C.

We wish to emphasize that the training data, prepared synthetically, was intended to mimic the snapshots of the LB simulation, and it is not generated by a physical process associated with LB simulations. However, it is noteworthy that the synthetic data preparation method described above bypasses the labor-intensive data acquisition part and minimizes human errors, thus making the development process of the droplet tracking application much quicker and more precise. This method can potentially be used in several other object detection applications. In the next section, we describe the YOLO network's training process to identify the droplets with the synthetic data.
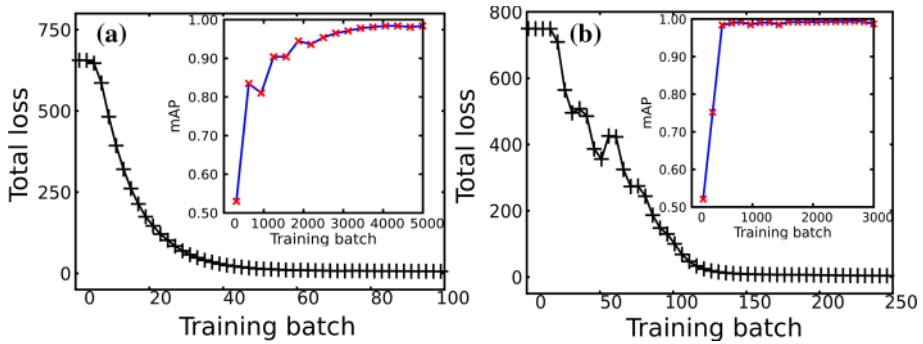
Fig. 6 Total loss and mean average precision (mAP) as the training progresses. **a** YOLO-tiny network, **b** YOLO network

3.4 Training

The training of the YOLO network is carried out through an iterative process. A subset of training data, called batch size, is passed through the network in each iteration, and then the network predicts the bounding boxes around the detected objects. A loss value is computed based on the difference between the output and the ground truth. The supplied label information is taken as the ground truth. The network parameters (weights and biases of the nodes) are updated to minimize the total loss and improve the accuracy of the network. For the YOLOv3 algorithm, the loss value, also called total loss, is the weighted sum of three separate contributions, regression loss, confidence loss, and classification loss [23]. In the training process, the weights of these contributions to the total loss are the same as proposed in Ref. [23] to optimize the mean average precision and maintain the model stability.

At regular intervals, i.e., after a few iterations, the accuracy of the network is assessed by computing mean average precision (mAP) [37,38]. The mean average precision is a good measure to know how well the trained network performs on a dataset that is different from the training dataset. Typically, a separate dataset, called validation dataset, is compiled to compute the mean average precision.

We trained two YOLO networks to identify the droplets from the LB simulations. We adapted the code for training the network from Ref. [39]. The first network, called YOLO-tiny, is a lighter version of the full YOLO network consisting of fewer convolutional layers than the full YOLO network to trade training and inference speed for accuracy. For many practical purposes, the lighter version (YOLO-tiny) does perform reasonably well.

For training the YOLO-tiny network, we consolidated a total of 10000 images, consisting of an equal number of two different types of synthetic images as shown in Fig. 14a,b. The parameters set for the training are mentioned here [40]. The parameters of the network are updated at the end of every batch. A separate dataset of 1000 images, called validation data, was prepared from the same type of images to calculate the mean average precision. The total loss and mean average precision as the training progresses are shown in Fig. 6a. As the training progresses, the total loss decreases, and the mean average precision of the network increases, indicating that the network is gaining performance with the training, reaching a total loss value very close to zero at the end of the training. The mean average precision value is saturated to 0.985, indicating that the network is 98.5% accurate in detecting the droplets in broad terms.

The second network we trained is the full YOLO network, with Darknet-53 as a backbone. Ten thousand images similar to the image shown in Fig. 14c were used as the training data. Another 1000 images of the same type were reserved as the validation data, and the training parameters are listed in [41]. In this case as well, as the training progresses, the total loss decreases and mean average precision augments (as seen in Fig. 6b) saturating to 0.99, i.e. the network is 99% accurate in detecting the droplets. The trained model in the form of a binary file containing the model's parameters (called weights) is saved. The weight file will be used later to analyze the data from LB simulations, and it is provided for both networks, the YOLO-tiny and the YOLO, as supplementary material [35].

By training the YOLO network, we accomplish the first task of droplet recognition from the digital images, while the DeepSORT algorithm handles the second task of droplet tracking. In the next section, we analyze two LB simulations as case studies to test and use the developed application.

4 Results

After the model is trained with the desired accuracy in terms of mean average precision, it is ready to analyze the real-world data. From simulations or experiments, the real word images can be given as input to the trained network, and individual droplets' trajectories can be obtained. This process is also called inference.

We run inference on the data generated with two LB simulations to test our trained network as described in Sect. 2. For inference, only the droplets (white color mass) were plotted on a dark background. The output video was then fed to the YOLO + DeepSORT network to infer the trajectories of individual droplets. We adapted code with Tensorflow implementation to run inference from Ref. [42].

4.1 Multi-core emulsions

Figure 7 shows a visual depiction of the object recognition by the YOLO network and tracking of the located objects by the DeepSORT network in four sequential images from the LB simulations. The bounding boxes predicted by the YOLO network are shown in blue color, and the unique ID assigned by the DeepSORT algorithm is written above each bounding box (see video3.avi).

It is then straightforward to extract trajectories of the individual droplets across sequential frames since we have dimensions of all the bounding boxes in all the frames with their unique ID. The center of mass of the individual droplets is approximated as the geometric center of its bounding box. Figure 8 shows the paths traced by the droplets.

4.2 Soft granular media

In the second case, we run inference on LB simulation of soft granular media. This time, we employed the YOLO network to identify the droplets since the trained YOLO-tiny network makes many mistakes in droplet recognition. This is likely due to the increased complexity of the system, which presents a densely packed configuration of droplets moving through a narrow channel significantly deviating from its initial shape.

The YOLO network, however, can locate the droplets with almost perfect accuracy. Here, we note again that the training of the YOLO network was carried out with the training data consisting of randomly placed dense clusters of solid white circles. As before, Fig. 9 shows
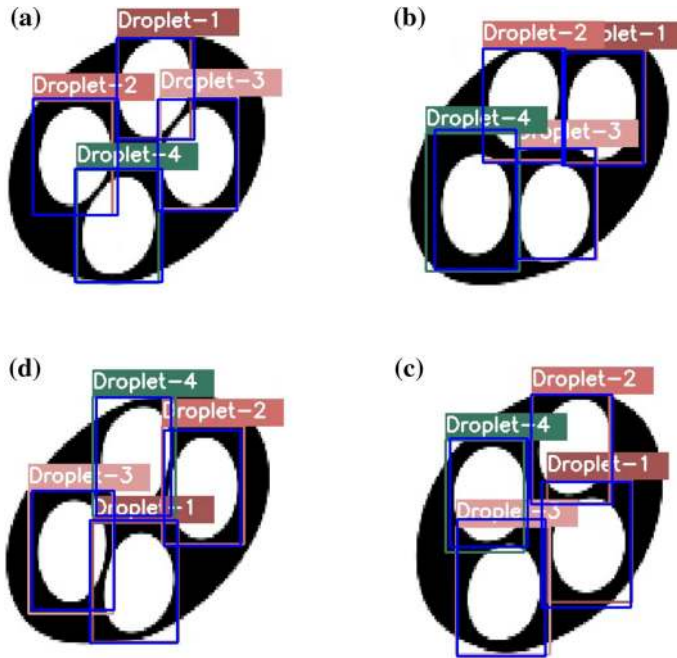
Fig. 7 Object detection and tracking at sequential frame number N. **a** $N = 18$, **b** $N = 19$, **c** $N = 20$, **d** $N = 21$
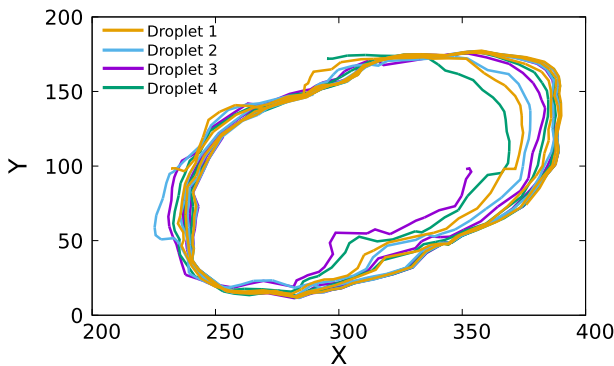

Fig. 8 Extracted trajectories of the center of mass of the individual droplets in a four core emulsion. On X-axis and Y-axis are the pixel numbers

the droplet identification by the YOLO network and tracking with the DeepSORT algorithm. The recognized droplets are shown with blue bounding boxes, and the unique IDs assigned by the DeepSORT algorithm are placed within the bounding box for visual inspection (see video4.avi).

In Fig. 10, we plot the extracted trajectories of the center of mass of the individual droplets. The center of the bounding box is approximated as the center of mass. In this case, we obtain 49 trajectories for the 49 droplets. Each separate trajectory is shown with a unique color.
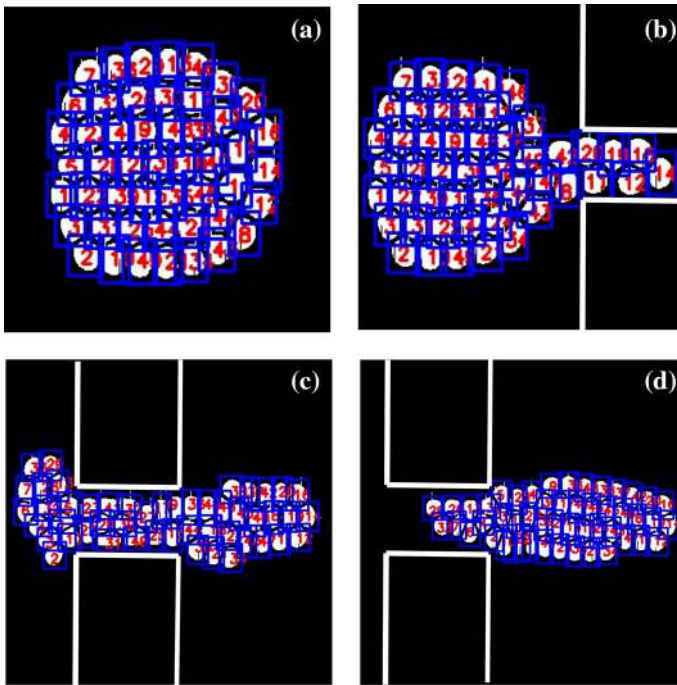
Fig. 9 Droplet recognition and tracking in four frames with frame number **a** $N = 5$, **b** $N = 6$, **c** $N = 8$, **d** $N = 9$. Blue bounding boxes are generated by the YOLO algorithm and unique IDs are assigned by the DeepSORT algorithm
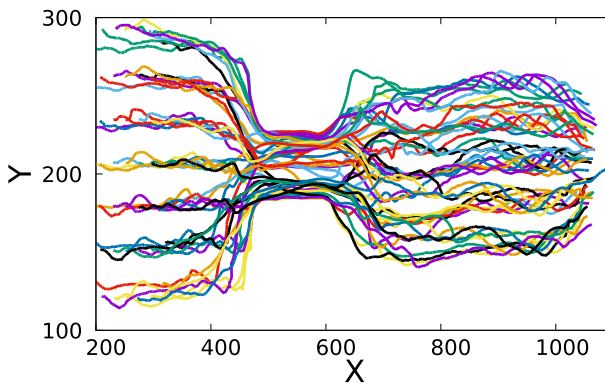

Fig. 10 Extracted trajectories of the center of mass of the individual droplets represented by unique colors. X and Y axis show the pixel number

4.3 Accuracy of the trained model

During the training, validation data were used to estimate the accuracy of the object detector model. In our case, they were statistically identical to the training dataset and could provide a biased estimate of the accuracy. Unbiased accuracy of the trained model can be obtained by analyzing a test case (also called test data), ideally from a real-world dataset with known
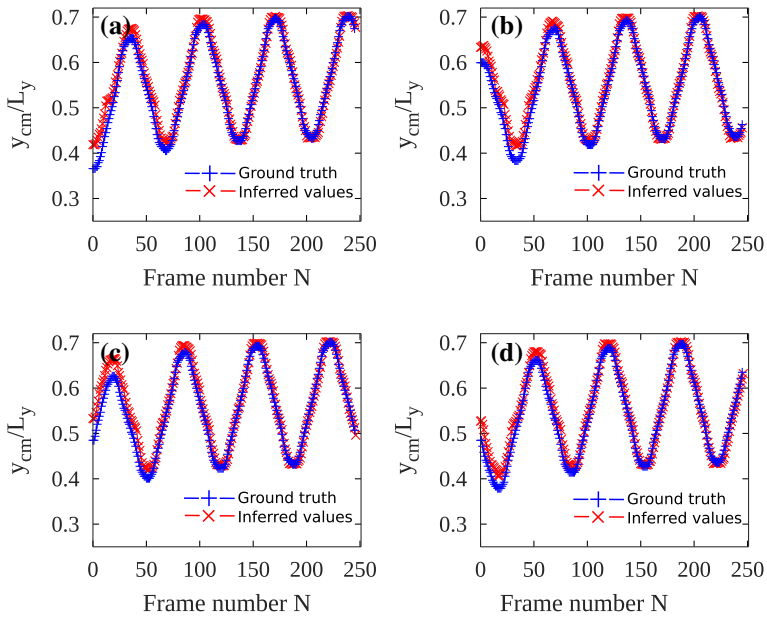
Fig. 11 a–d Comparison of the inferred center of mass of the four individual droplets in every frame with the ground truth computed by the analytical expression in Ref. [34]. Mean square error MSE between the inferred values and the ground truth values is computed to measure the error. (**a**) $MSE = 3.44 \times 10^{-2}$, (**b**) $MSE = 1.80 \times 10^{-2}$, (**c**) $MSE = 3.12 \times 10^{-2}$, (**d**) $MSE = 1.59 \times 10^{-2}$

droplet trajectories. From the output trajectories of the model and the known trajectories as ground truth, the accuracy of the trained model can be estimated for real-world applications.

To measure the accuracy of the trained model, we compute mean square errors (MSE) by a Euclidean metric:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_{cm}(i) - \hat{y}_{cm}(i))^2. \tag{1}$$

Here, N is the total number of frames, y_{cm} is the ground truth y coordinate of the center of mass, $\hat{y}_{cm}$ is the inferred y coordinate of the center of mass by the deep learning tool.

We measure the accuracy of the YOLO-tiny + DeepSORT model by comparing the inferred trajectories of the droplets with the known ground truth. In this case, the ground truth is the center of mass computed with an analytical expression (see Ref. [34]). In Fig. 11, we compare the ground truth y-coordinates of the center of mass (y_{cm}) of individual droplets with the inferred y-coordinate values ($\hat{y}_{cm}$) by the deep learning tool. We observe that the deep learning model agrees with the analytical expression after a few initial frames. However, in the initial frames, the deep learning model slightly deviates from the true values, even though both the curves are in qualitative agreement. The mean square value MSE is of the order of 10^{-2} for all four tracked droplets, indicating a good agreement between the ground truth and the inferred values.

For measuring the accuracy of the full YOLO + DeepSORT model, we prepared a synthetic test case where the center of mass of the droplets move along some predefined paths (see Appendix D). The predefined paths serve as the ground truth. We employed the YOLO + DeepSORT model to infer trajectories of the droplets from the video and the comparison is
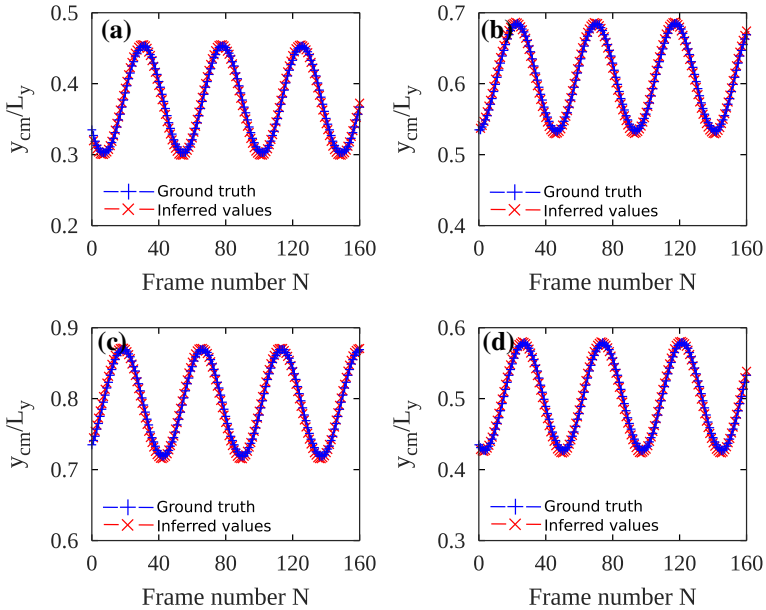
Fig. 12 Measuring accuracy of the YOLO + DeepSORT model. **a**–**d** Inferred path of four different droplets compared with their predefined paths. Mean square error MSE between the inferred values and the ground truth values is computed to measure the error. (**a**) $MSE = 1.41 \times 10^{-4}$, (**b**) $MSE = 1.52 \times 10^{-4}$, (**c**) $MSE = 1.55 \times 10^{-4}$, (**d**) $MSE = 1.45 \times 10^{-4}$

shown in Fig. 12. Four droplets were tracked and the inferred path is compared with the known predefined trajectories of these droplets. Both the trajectories are in excellent agreement and we see no deviation in the inferred and true trajectories in any phase of the tracking. The YOLO + DeepSORT model is highly accurate with MSE of the order of 10^{-4}. The low values of mean square errors once again point to the high accuracy of the trained model.

We emphasize that we measure the accuracy of the developed application, including the object recognition and the object tracking process. The error values are indicative of the total expected error in extracted trajectories from the real-world data. Apart from the model's accuracy, the model's analysis speed is an important factor to consider while developing an application for practical use. In the next section, we report the inference speed, i.e., the rate of image analysis by the developed application on different computer hardware configurations.

4.4 Inference speed

We test the inference speed of the object detection and tracking model on a typical notebook computer and a commonly used GPU machine. For inference, we used Tensorflow implementation [42]. The inference speed, analyzed frame per second (FPS), for the YOLO-tiny and the YOLO network are tabulated in Table 1.

These data clearly show that there is room for improvement and optimization of the inference speed for both YOLO and DeepSORT, for instance, by coding them in other programming languages, such as C++. This is an interesting topic for future work.

Having accurate trajectories of moving objects is the first step to study the system. In the next section, we present an analysis of the inferred trajectories to test a hypothesis.

Table 1 Average inference speed in frames per second (fps) on two hardware configurations

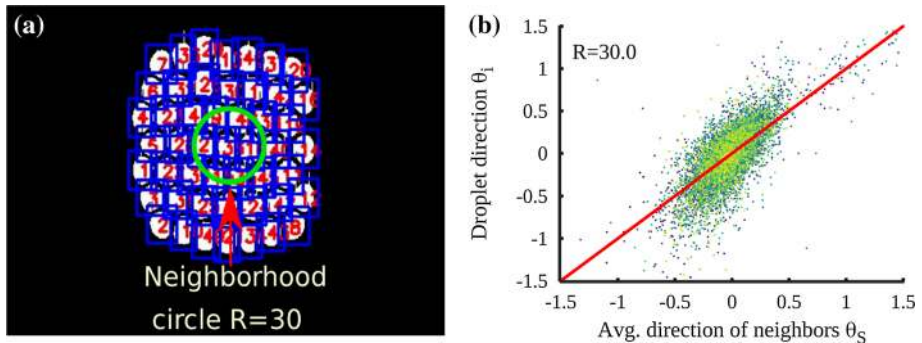| Network/Machine	Inference speed	
	CPU (i3-2328M)	GPU (RTX 2060 Super)
YOLO-tiny + DeepSORT	2.44 fps	35 fps
YOLO + DeepSORT	0.25 fps	12 fps

Fig. 13 **a** Neighborhood of one of the randomly chosen droplet. The neighborhood of a droplet is a circle with $R = 30$ pixels. **b** Scatter plot of the direction of individual droplets θ_i with the average moving direction of droplets θ_S in the neighbourhood of the droplet i. Values on the axes are in radians. The thick red line is a guideline to identify instances when the moving direction of droplets and their neighbors are identical

4.5 Flocks of droplets?

At first sight, the droplets seem to move like the agents in active matter systems, such as birds in a flock. Several statistical physics models have been proposed to explain the flocking behavior in animal groups [43–45], and experimental studies involving observation of moving animal groups inferred the interaction rules between the agents [46,47]. One prominent feature common in most of these models is that the agents (birds) move along the average direction of their neighbors.

Given the superficial similarity between a flock and the soft granular media system studied here, we perform a quick analysis to test a hypothesis that the droplets behave like agents in an active matter system. To this end, we calculate the average moving direction θ_S of the neighbors of each individual droplet. The neighbors of droplet i are defined as droplets within a circle of radius R, fixed on the center of the droplet i. We set the size of the neighborhood in such a way that few neighboring droplets are included (see Fig. 13a). Figure 13b shows a scatter plot between individual droplet's moving direction θ_i and the average direction of their neighbors θ_S for the entire duration of the simulation. The moving direction is measured as angle θ made by the velocity vector with the x-axis in a fixed frame of reference. Thus, the range of θ is from $-\pi$ to $+\pi$. Each different color in this plot shows a separate identity of the individual droplets. The red line is a guideline for the eyes to show the situation when the individual droplet's moving direction is identical to that of its neighbors.

In Fig. 13, around 47% of all the total points are in the band when $|\theta_i - \theta_S| < 0.12$. i.e., when the angle between the droplet's motion and its neighbors is less than 7 degrees. This suggests that the moving direction of individual droplet deviates (by more than 7 degrees) from the average direction of their neighbors 53% of the time.

We conclude that individual droplets frequently deviate from the average direction of their neighbors than the individual birds do in a polarized flock. This is plausible since, besides some similarities, there are also several points of departure between the two systems. First, in soft granular media systems, the fluid flow carries the droplets along the flow direction, like a systematic wind and second, the wall boundaries impose a stringent constraint on the motion of the droplets. None of the two aspects above are generally considered in the modeling of active matter systems.

In this analysis, we compared only the polarization between the system of droplets and the ordered flock. Numerous other observations characterize a flock [48], and a detailed comparison is required to quantify the similarities between these two systems. Nevertheless, this raises an interesting question on how to generalize the flock equations to come up with a set of effective equations of the droplets in complex many-body flows such as the one explored here. We plan to address this question in future work.

5 Conclusions

We adapted and combined deep learning-based object recognition algorithm YOLO and object tracking algorithm DeepSORT to analyze data generated by fluid simulations of two soft flowing systems, multicore emulsions, and soft granular media. In particular, the trajectories of the individual droplets moving within a microchannel flow were extracted from the digital images of the system.

We determine the accuracy of the complete application by comparing the inferred trajectories with the ground truth trajectories computed by independent methods. Although we used LB simulation data as a case study; the developed application can also be applied data generated by two-dimensional experimental setups.

With the use of commonly available GPUs, the YOLO + DeepSORT based application can analyze the images at a remarkable rate, above the image capture rate of typical cameras, thus offering a practical, low-cost application capable of analyzing real-time data as it is acquired.

From the technical point of view, we showed that the synthetically prepared datasets could be used to train the object detection network with almost perfect accuracy, thus avoiding labor-intensive, time-consuming training data acquisition for object recognition network training.

We combined two state-of-the-art algorithms to identify and track the droplets and tested the model on two physical systems. It would be interesting to see how the application works for other dense emulsion systems. Nevertheless, YOLO and DeepSORT have shown remarkable performance in object recognition and tracking in more complex environments. We hope and expect that further developments of the work presented here may pave the way to the automatic detection and tracking of moving agents in complex flows of scientific, engineering, and biological interest.

Data Availability Statement This manuscript has associated data in a data repository. [Authors comment: All data generated or analysed during this study are included in this published article.]

A Color-gradient approach with near contact interactions

Here we describe the LB approach used to simulate the translocation through a constriction of a high internal phase double emulsion. We employ a color-gradient [49–54] regularized LB method [55–57], which is built starting from two sets of probability distribution functions capturing the dynamics of the immiscible fluid components. Each set evolves via a sequence of streaming and collision steps [30,58–60]

$$f_i^k(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i^k(\mathbf{x}, t) + \Omega_i^k(f_i^k(\mathbf{x}, t)) + F_i^{rep}, \tag{2}$$

where $f_i^k(\mathbf{x}, t)$ is the ith discrete probability distribution function for the kth component, giving the probability of finding a fluid particle at position $\mathbf{x}$, time t and with discrete velocity $\mathbf{c}_i$. The index k is such that $k = 1, 2$, while i belongs to the range $0 \le i \le N_{set}$, where N_{set} is the dimension of the set of discrete probability distribution functions and is equal to 8 for the two-dimensional nine speeds lattice (D2Q9) employed in this paper. The time step Δt is expressed in lattice units [30] and set to 1, a usual choice in the LB methods [58]. Finally, F_i^{rep} is a force aimed at upscaling the repulsive near-contact forces acting on scales much smaller than the resolved ones [61,62].

As stated above, the multicomponent approach employed in this work is based on a variant of the color-gradient LB model. Following the standard formalism, the collision operator can be split into three parts

$$\Omega_i^k = (\Omega_i^k)^{(3)}[(\Omega_i^k)^{(1)} + (\Omega_i^k)^{(2)}]. \tag{3}$$

The first term $(\Omega_i^k)^{(1)}$ is the usual single relaxation time Bhatnagar–Gross–Krook collisional operator [58] while the second part $(\Omega_i^k)^{(2)}$ is the *perturbation step*, which accounts for the interfacial tension, and reads as

$$(\Omega_i^k)^{(2)} = \frac{A_k}{2} |\nabla \Theta| \left(w_i \left(\frac{\mathbf{c}_i \cdot \nabla \Theta}{|\nabla \Theta|} \right)^2 - B_i \right), \tag{4}$$

where A_k ($k = 1, 2$) and B_i ($i = 0, N_{set}$) are suitable constants defined in [61]. In Eq. (4), Θ is a scalar phase field, defined as

$$\Theta = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} \tag{5}$$

assuming the value 1 in the fluid component with density ρ_1 and the value -1 in the fluid component with density ρ_2. It is worth observing that the constants A_1, A_2 are related to the surface tension σ by

$$\sigma = \frac{2}{9} \tau (A_1 + A_2). \tag{6}$$

The third term $(\Omega_i^k)^{(3)}$ is the recolouring operator, as described in [63], which aims at minimizing the mutual diffusion between the fluid components, thus favouring their separation.

Finally, the last term at the right hand side of Eq. (2) codes for the short-range, repulsive force at the interface between the two fluid components, aimed at frustrating the coalescence between interacting, neighboring droplets.

B Free energy LB models for interacting multicomponent fluids

In this subsection we shortly outline the multicomponent field model for immiscible fluid mixtures employed to simulate the physics of three and four-core emulsions. Further details can be found in [33,34,64–66].

In this approach, a set of scalar phase-fields $\psi_i(\mathbf{r}, t)$, $i = 1, ..., M$ (where M is the number of cores) is used to model the droplet density (positive within each drop and zero outside), while the average fluid velocity of both drops and solvent is described by a vector field $\mathbf{v}(\mathbf{r}, t)$.

The dynamics of the fields $\psi_i(\mathbf{r}, t)$ is governed by a set of convection-diffusion equations

$$\partial_t \psi_i + \mathbf{v} \cdot \nabla \psi_i = D \nabla^2 \mu_i, \tag{7}$$

where D is the mobility, $\mu_i = \frac{\delta \mathcal{F}}{\delta \psi_i}$ is the chemical potential and $\mathcal{F} = \int_V f \, dV$ is the total free energy describing the equilibrium properties of the fluid suspension. The free energy density f is given by [67,68]

$$f = \frac{a}{4} \sum_i^M \psi_i^2 (\psi_i - \psi_0)^2 + \frac{k}{2} \sum_i^M (\nabla \psi_i)^2 + \epsilon \sum_{i,j,i<j} \psi_i \psi_j, \tag{8}$$

where the first term is the double-well potential ensuring the existence of two coexisting minima, $\psi_i = \psi_0$ inside the ith droplet and $\psi_i = 0$ outside, and the second term stabilizes the droplet interface. The parameters a and k are two positive constants related to the surface tension $\sigma = \sqrt{8ak/9}$ and the width of interface $\xi = 2\sqrt{2k/a}$ [30,69]. Finally, the last term in Eq. (8) is a soft-core repulsion contribution which penalizes the overlap of droplets, and whose strength is tuned by the positive constant ϵ.

The fluid velocity $\mathbf{v}$ obeys the Navier-Stokes equation, which in the incompressible limit reads

$$\rho \left(\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \right) \mathbf{v} = -\nabla p + \eta \nabla^2 \mathbf{v} - \sum_i \psi_i \nabla \mu_i. \tag{9}$$

Here ρ is the fluid density, p is the isotropic pressure and η is the dynamic viscosity. A typical set of values of thermodynamic parameters used in simulations is the following: $a = 0.07$, $k = 0.1$, $D = 0.1$, $\eta = 1.67$, $\epsilon = 0.05$, $\Delta x = 1$ (lattice step), $\Delta t = 1$ (time step).

Equations (7) are solved using a finite difference scheme while Eq. (9) is integrated by means of a standard lattice Boltzmann method. The latter shares many features with the LB model described in the previous section. Indeed, it is built from a set of distribution functions whose evolution is governed by a discrete Boltzmann equation akin to Eq. (2) and defined on a D2Q9 lattice. In both methods it can be shown that, once conservation of mass and momentum are fulfilled, the Navier-Stokes equation can be obtained by performing a Chapman-Enskog expansion of the distribution functions [58,70]. However, unlike the previous algorithm, in this approach the thermodynamics is encoded in a free-energy employed to calculate the forces (chemical potential and pressure tensor) controlling the relaxation dynamics of the mixture. In addition, this model allows for an immediate tracking of position and speed of the droplets by computing the coordinates of the center of mass.

C Training data

Training data is used to train the YOLO network to locate the droplets in given input images. Now we describe the training data acquisition employed in this work. Training data for the YOLO network must consist of several images of the objects of interest and their positions in the images. Training data is passed through the network, and the network parameters are updated with every pass to increase the detection accuracy. In a typical exercise to train a network, training data is compiled by acquiring images taken by digital cameras in the real world. Objects in those images are manually identified and marked with bounding boxes encapsulating each object. A separate label file containing the class of the object, positions (x,y coordinates), and dimensions (width and height) of the bounding boxes is prepared.

We avoided the labor-intensive part of image acquisition and manual labeling by preparing a synthetic training dataset. The training dataset must contain several images that bear some visual features of the expected outcome of the LB simulations, as shown in Figs. 1 and 2. In our case, the outcome of the LB simulations are the images with the area occupied by the droplets shown by white color on a high contrast dark background. We prepared our training dataset by randomly placing solid white circles on a dark uniform background. The solid white circles were intended to resemble the droplets, and the high contrast background is intended for easy identification of the droplets from the background.

In the case of LB simulation of multi-core emulsions, the outcome consists of few elliptical shape droplets enclosed in a dark bag on a uniform white background. We incorporate these features in the training data set by preparing 10000 images each of two types of images. The first type of images (see a representative Fig. 14a) are prepared by placing the white solid circles and ellipses of randomly chosen semi-minor and semi-major axis on the background of size 1200 × 600 to teach the network to identify droplets of various sizes. In the second type of images (see Fig. 14b), two solid circles as droplets are placed randomly within a solid black circle. The position of the solid black circle is chosen randomly in the uniform white background.
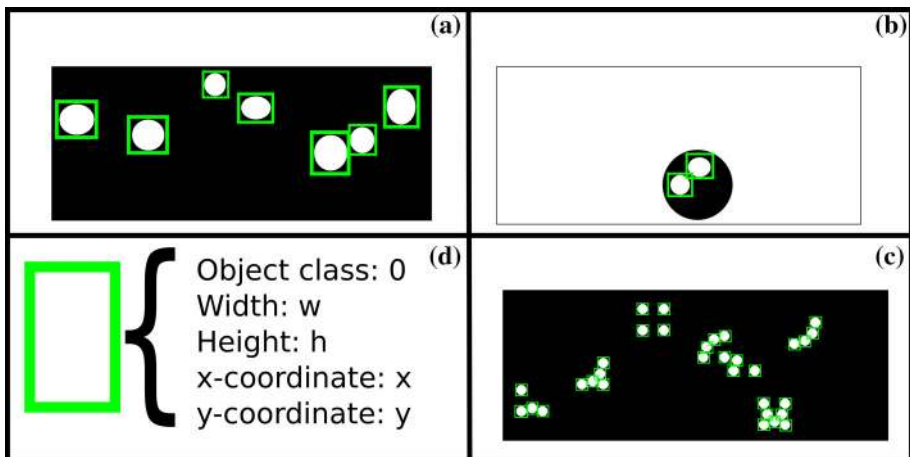

Fig. 14 **a** Randomly placed droplets on a black uniform background of size 1200 × 600. **b** Two droplets are randomly placed in a black envelope. The placement of the envelope is random within the white uniform background of size 1200 × 600 pixels. **c** Few random size clusters of droplets were placed randomly in a uniform background of size 1200 × 600. **d** Label information associated with each of the bounding boxes

In the case of LB simulations of soft granular media, the most apparent feature is the dense droplets cluster. In this case, few clusters with the random number of solid white circles are placed in a uniform solid black background to compile training data. The training data consisted of 10000 such images (see Fig. 14c as an example).

The label information for every image is shown in Fig. 14d. Each bounding box encloses a single white droplet. In associated text files, the dimensions of the bounding boxes (x and y coordinates, width and height, and the class of object it encapsulates) are noted down. Images and the training data together form a training dataset for the YOLO network.

D Synthetic test case

To measure the accuracy of the developed YOLO + DeepSORT application, we prepared a test video. In this video, the center of mass of each droplet move along predefined trajectories (see video5.avi). The following equation gives the trajectory of individual droplets;

$$
\begin{aligned}
x^i(t+1) &= x^i(t) + A, \\
y^i(t+1) &= y^i(t) + B\ sin(kx^i(t)).
\end{aligned}
\tag{10}
$$

Here, $x^i(t)$ and $y^i(t)$ are the coordinates of the ith droplet at time t. A, B and k are the suitable constants set as $A = 5$, $B = 5$ and $k = 5$.

In practice, we prepared several images and in each image we pasted 12 droplets at the location given by Eq. (10). Each droplet was placed with a random initial location, i.e. at pixel location given as $(y^i(t = 0), x^i(t = 0))$. These images were then sequentially stacked to prepare a video of continuously moving droplets on predefined paths. In Fig. 15 we show one instance of the droplet identification and tracking process. See video6.avi for the output of the developed application.
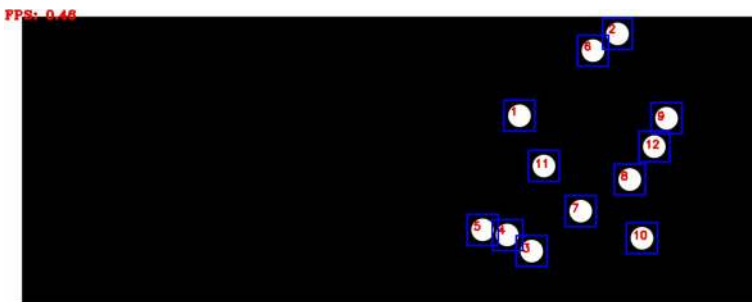

Fig. 15 Snapshot of the output from YOLO + DeepSORT network to track droplets moving on predefined paths. Blue boxes are predictions by the YOLO network, and the DeepSORT network assigned unique ids

References

1. S. Succi, P.V. Coveney, Big data: the end of the scientific method? Philosoph. Transac. Royal Soc. A: Math. Phys. Eng. Sci. **377**(2142), 20180145 (2019)
2. C. Rudin, K.L. Wagstaff, Machine learning for science and society. Mach. Learn. **95**, 1–9 (2014)
3. M.G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L.H. Leufen, A. Mozaffari, S. Stadtler, Can deep learning beat numerical weather prediction? Philosoph. Transac. Royal Soc. A Mathe. Phys. Eng. Sci. **379**(2194), 20200097 (2021)
4. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**, 436–444 (2015)
5. Darmatasia, M. I. Fanany, Handwriting recognition on form document using convolutional neural network and support vector machines (cnn-svm), 2017 5th International Conference on Information and Communication Technology (ICoIC7), pp. 1–6, (2017)
6. S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, B. Yoon, Improved handwritten digit recognition using convolutional neural networks (cnn), Sensors, **20**(12), (2020)
7. N. H. Tandel, H. B. Prajapati, V. K. Dabhi, Voice recognition and voice comparison using machine learning techniques: A survey, 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 459–465, (2020)
8. K. Han, D. Yu, I. Tashev, Speech emotion recognition using deep neural network and extreme learning machine, in Interspeech 2014, (2014)
9. A. Severyn and A. Moschitti, Unitn: Training deep convolutional neural network for twitter sentiment classification, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 464–469, (2015)
10. A. M. Ramadhani, H. S. Goo, Twitter sentiment analysis using deep learning methods, 2017 7th International Annual Engineering Seminar (InAES), pp. 1–4, (2017)
11. L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: a survey. WIREs Data Min. Knowl. Discovery **8**(4), e1253 (2018)
12. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences. Rev. Mod. Phys. **91**, 045002 (2019)
13. A.W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A.W.R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D.T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, Improved protein structure prediction using potentials from deep learning. Nature **577**, 706–710 (2020)
14. J.W. Khor, N. Jean, E.S. Luxenberg, S. Ermon, S.K.Y. Tang, Using machine learning to discover shape descriptors for predicting emulsion stability in a microfluidic channel. Soft Matt. **15**, 1361–1372 (2019)
15. Y. Mahdi, K. Daoud, Microdroplet size prediction in microfluidic systems via artificial neural network modeling for water-in-oil emulsion formulation. J. Dispersion Sci. Technol. **38**(10), 1501–1508 (2017)
16. T. Osman, S. S. Psyche, J. M. Shafi Ferdous, H. U. Zaman, Intelligent traffic management system for cross section of roads using computer vision, 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1–7, (2017)
17. G. T. S. Ho, Y. P. Tsang, C. H. Wu, W. H. Wong, K. L. Choy, A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities, Sensors, **19**(8), (2019)
18. B. Yogameena, C. Nagananthini, Computer vision based crowd disaster avoidance system: a survey. Int J Disaster Risk Reduction **22**, 95–129 (2017)
19. N. Ragesh, B. Giridhar, D. Lingeshwaran, P. Siddharth, K. P. Peeyush, Deep learning based automated billing cart, 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0779–0782, (2019)
20. R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, J. M. Z. Maningo, Object detection using convolutional neural networks, TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 2023–2027, (2018)
21. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transac Pattern Analy Machine Intell **39**(6), 1137–1149 (2017)
22. N. Wojke, A. Bewley, D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649, (2017)
23. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, (2016)
24. J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," ArXiv:1804.02767v1, (2018)
25. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector. Comput. Vision - ECCV **2016**, 21–37 (2016)
26. N. S. Punn, S. K. Sonbhadra, S. Agarwal, G. Rai, Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques (2021). arXiv:2005.01385v4 [cs.CV]

27. S. Khosravipour, E. Taghvaei, N. M. Charkari, Covid-19 personal protective equipment detection using real-time deep learning methods (2021). arXiv:2103.14878v1 [cs.CV]
28. T. Zhang, X. Zhang, Y. Yang, Z. Wang, G. Wang, Efficient golf ball detection and tracking based on convolutional neural networks and kalman filter (2021). arXiv:2012.09393v2 [cs.CV]
29. K. Host, M. Ivašić-Kos, M. Pobar, "Tracking handball players with the deepsort algorithm," in Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods - ICPRAM, pp. 593–599, INSTICC, SciTePress, (2020)
30. T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E.M. Viggen, *The lattice boltzmann method* (Springer, Newyork, 2017)
31. M. Costantini, C. Colosi, J. Guzowski, A. Barbetta, J. Jaroszewicz, W. Swieszkowski, M. Dentini, P. Garstecki, Highly ordered and tunable polyhipes by using microfluidics. J. Mater. Chem. B **2**, 2290–2300 (2014)
32. A.S. Utada, E.L. Lorenceau, D.R. Link, P.D. Kaplan, H.A. Stone, D.A. Weitz, Monodisperse double emulsions generated from a microcapillary device. Science **308**, 537–541 (2005)
33. A. Tiribocchi, A. Montessori, S. Aime, M. Milani, M. Lauricella, S. Succi, D.A. Weitz, Novel nonequilibrium steady states in multiple emulsions. Phys. Fluids **32**, 017102 (2020)
34. A. Tiribocchi, A. Montessori, F. Bonaccorso, M. Lauricella, S. Succi, Concentrated phase emulsion with multicore morphology under shear: a numerical study. Phys. Rev. Fluids **5**, 113606 (2020)
35. https://cloud.degoo.com/share/R_EZJd7UptbslTVcuV6Rqg ,
36. M. Durve, F. Bonaccorso, A. Montessori, M. Lauricella, A. Tiribocchi, S. Succi, A fast and efficient deep learning procedure for tracking droplet motion in dense microfluidic emulsions (2021). arXiv:2103.01572v1 [cond-mat.soft]
37. M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge. Int. J. Comput. Vision **88**, 303–338 (2010)
38. P. Henderson, V. Ferrari, "End-to-end training of object class detectors for mean average precision," Computer Vision – ACCV 2016 Lecture Notes in Computer Science, 198–213, (2017)
39. J. Redmon, "Darknet: Open source neural networks in c." http://pjreddie.com/darknet/, year, 2013–2016
40. Batches=10000, batch size=128, learning rate=0.001
41. Batches=10000, batch size=64, learning rate=0.001
42. https://github.com/theAIGuysCode/yolov3_deepsort
43. T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. **75**, 1226–1229 (1995)
44. I.D. Couzin, J. Krause, R. James, G.D. Ruxton, N.R. Franks, Collective memory and spatial sorting in animal groups. J. Theor. Biol. **218**(1), 1–11 (2002)
45. A. Cavagna, L. Del Castello, I. Giardina, T. Grigera, A. Jelic, S. Melillo, T. Mora, L. Parisi, E. Silvestri, M. Viale, A.M. Walczak, Flocking and turning: a new model for self-organized collective motion. J. Stat. Phys. **158**, 601–627 (2015)
46. J.E. Herbert-Read, A. Perna, R.P. Mann, T.M. Schaerf, D.J.T. Sumpter, A.J.W. Ward, Inferring the rules of interaction of shoaling fish. Proc. Nat. Acad. Sci. **108**(46), 18726–18731 (2011)
47. R. Lukeman, Y.-X. Li, L. Edelstein-Keshet, Inferring individual rules from collective behavior. Proc. Nat. Acad. Sci. **107**(28), 12576–12580 (2010)
48. I. Giardina, Collective behavior in animal groups: theoretical models and empirical studies. HFSP Journal **2**, 205–219 (2008)
49. S. Leclaire, M. Reggio, J.-Y. Trépanier, Numerical evaluation of two recoloring operators for an immiscible two-phase flow lattice boltzmann model. Appl. Math. Model. **36**(5), 2237–2252 (2012)
50. S. Leclaire, A. Parmigiani, O. Malaspinas, B. Chopard, J. Latt, Generalized three-dimensional lattice boltzmann color-gradient method for immiscible two-phase pore-scale imbibition and drainage in porous media. Phys. Rev. E **95**, 033306 (2017)
51. A.K. Gunstensen, D.H. Rothman, S. Zaleski, G. Zanetti, Lattice boltzmann model of immiscible fluids. Phys. Rev. A **43**, 4320 (1991)
52. A. Montessori, M. Lauricella, M. La Rocca, S. Succi, E. Stolovicki, R. Ziblat, D. Weitz, Regularized lattice boltzmann multicomponent models for low capillary and reynolds microfluidics flows. Comput. Fluids **167**, 33–39 (2018)
53. A. Montessori, M. Lauricella, S. Succi, E. Stolovicki, D. Weitz, Elucidating the mechanism of step emulsification. Phys. Rev. F. **3**, 072202 (2018)
54. D.H. Rothman, J.M. Keller, Immiscible cellular-automaton fluids. J. Stat. Phys. **52**(3–4), 1119–1127 (1988)
55. A. Montessori, P. Prestininzi, M. La Rocca, S. Succi, Lattice boltzmann approach for complex nonequilibrium flows. Phys. Rev. E **92**(4), 043308 (2015)

56. J. Latt, B. Chopard, Lattice boltzmann method with regularized pre-collision distribution functions. Math. Comput. Simul. **72**(2–6), 165–168 (2006)
57. C. Coreixas, B. Chopard, J. Latt, Comprehensive comparison of collision models in the lattice boltzmann framework: theoretical investigations. Phys. Rev. E **100**(3), 033305 (2019)
58. S. Succi, *The Lattice Boltzmann equation: for complex states of flowing matter* (Oxford University Press, London, 2018)
59. S. Succi, Lattice boltzmann 2038. EPL (Europhys. Lett.) **109**(5), 50001 (2015)
60. R. Benzi, S. Succi, M. Vergassola, The lattice boltzmann equation: theory and applications. Phys. Rep. **222**(3), 145–197 (1992)
61. A. Montessori, M. Lauricella, N. Tirelli, S. Succi, Mesoscale modelling of near-contact interactions for complex flowing interfaces. J. Fluid Mech. **872**, 327–347 (2019)
62. A. Montessori, M. Lauricella, A. Tiribocchi, S. Succi, Modeling pattern formation in soft flowing crystals. Phys. Rev. Fluids **4**(7), 072201 (2019)
63. M. Latva-Kokko, D.H. Rothman, Diffusion properties of gradient-based lattice boltzmann models of immiscible fluids. Phys. Rev. E **71**(5), 056702 (2005)
64. M. Foglino, A.N. Morozov, O. Henrich, D. Marenduzzo, Flow of deformable droplets: discontinuous shear thinning and velocity oscillations. Phys. Rev. Lett. **119**, 208002 (2017)
65. M. Foglino, A.N. Morozov, D. Marenduzzo, Rheology and microrheology of deformable droplet suspensions. Soft Matter **14**, 9361–9367 (2018)
66. A. Tiribocchi, A. Montessori, M. Lauricella, F. Bonaccorso, S. Succi, S. Aime, M. Milani, D.A. Weitz, The vortex-driven dynamics of droplets within droplets. Nat. Commun. **12**, 82 (2021)
67. S.R. De Groot, P. Mazur, *Non-equilibrium thermodynamics* (NY, Dover, New York, 1984)
68. G. Lebon, D. Jou, J. Casas-Vazques, *Understanding non-equilibrium thermodynamics: foundations, applications* (Springer, Frontiers, 2008)
69. J.S. Rowlinson, B. Widom, *Molecular theory of capillarity* (Clarendon Press, Oxford, 1982)
70. L.N. Carenza, G. Gonnella, A. Lamura, G. Negro, A. Tiribocchi, Lattice boltzmann methods and active fluids. Eur. Phys. J. E **42**, 81 (2019)