Machine Learning with Applications 7 (2022) 100222

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Microfluidic droplet detection via region-based and single-pass convolutional neural networks with comparison to conventional image analysis methodologies

Gregory Philip Rutkowski [a,*], Ilgar Azizov [a], Evan Unmann [b], Marcin Dudek [a], Brian Arthur Grimes [a]

[a] *Ugelstad Laboratory, Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway*
[b] *Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC, USA*

ARTICLE INFO

Keywords:
Droplet microfluidics
Emulsion
Image analysis
Object detection
Convolutional neural network
Deep learning

ABSTRACT

As the complexity of microfluidic experiments and the associated image data volumes scale, traditional feature extraction approaches begin to struggle at both detection and analysis pipeline throughput. Deep-neural networks trained to detect certain objects are rapidly emerging as data gathering tools that can either match or outperform the analysis capabilities of the conventional methods used in microfluidic emulsion science. We demonstrate that two types of neural-networks, You Only Look Once (YOLOv3, YOLOv5) and Faster R-CNN, can be trained on a dataset which comprises of droplets generated across several microfluidic experiments and systems. The latitude of droplets used for training and validation, produce model weights which are easily transitive to emulsion systems at large, while completely circumventing any necessity of manual feature extraction. In flow cell experiments which comprised of greater than either 10,000 mono- or polydisperse droplets, the models show excellent or superior statistical symmetry to classical implementations of the Hough transform or widely utilized ImageJ plugins. In more complex chip architectures which simulate porous media, the produced image data typically requires heavy pre-processing to extrapolate valid data, where the models were able to handle raw input and produce size distributions with accuracy of ± 2 μm for intermediate magnifications. This data harvesting fidelity is extended to foreign datasets not included in the training such as micrograph observation of various emulsified systems. Implementing these neural networks as the sole feature extraction tools in these microfluidic systems not only makes the data pipelining more efficient but opens the door for live detection and development of autonomous microfluidic experimental platforms due to inference times of greater than 100 frames per second.

1. Introduction

Emulsions are kinetically-stabilized systems of certain liquids dispersed in others. Their stability and destabilization mechanisms are a crucial factor in a number of everyday products and industrial processes, such as food products, cosmetics, production of pharmaceuticals and wastewater treatment (Hiemenz & Rajagopalan, 1997). Destabilization most commonly occurs through coalescence and gravity separation, however other physicochemical processes such as Ostwald ripening or flocculation between droplets can also contribute to phase separation. Based on these stability processes, the continued research regarding measurement techniques which probe these phenomena is vital for ensuring development of the aforementioned systems and products. The physical tools which contribute to these techniques include microscopy, turbidity measurements, light scattering, rheology, droplet manipulators, and more recently, microfluidics.

Within emulsion science, microfluidics is a new, but unsurprising addition to the experimental toolbox. Droplet-based microfluidics is a perfectly suited technique for studying emulsified systems due to its ability to precisely tailor and manipulate droplet parameters (Anna, 2016). As far back as 2012, Bremond and Bibette (Bremond & Bibette, 2012) were able to identify microfluidics as a powerful tool for approaching emulsion science research (based on the steadily increasing number of reports produced yearly). Importantly, it enables work to be performed within a similar size scale common for industrially-produced emulsions, but simultaneously reduces throughput volumes, thus significantly reducing waste produced during measurements. Furthermore,

[*] Corresponding author.
E-mail addresses: gregory.p.rutkowski@ntnu.no (G.P. Rutkowski), ilgar.azizov@ntnu.no (I. Azizov), eunmann@uncc.edu (E. Unmann), marcin.dudek@ntnu.no (M. Dudek), brian.grimes@ntnu.no (B.A. Grimes).

https://doi.org/10.1016/j.mlwa.2021.100222
Received 30 August 2021; Received in revised form 18 October 2021; Accepted 13 November 2021
Available online 27 November 2021

the inherent transparency of the microfluidic devices used in these studies facilitates visual observation of these phenomena, often via high-speed imaging. Therefore, classical image analysis has become a tandem subdiscipline that accounts for much of the harvestable data in these emulsified systems (Shang et al., 2017).

The inherent size of the flow channels in these microfluidic chips or devices (<1 mm), coupled to the velocity and number of droplets within an active system poses a challenge in regards to the technical specifications for visualization. Compared to steady-state microscopy, capturing the generation, flow and behavior of hundreds or thousands of droplets per second often requires a high-speed camera operating in unison with the optics of a microscope. The definition which constitutes the baseline requirement for "high-speed imaging" is often vague and depends on the application, however in most cases it starts at the higher limit for standard microscopy imaging, usually 50–100 frames per second (fps). Most commercial high-speed cameras excel at recording greater than 1000 fps (Versluis, 2013), and indeed we see that many reports within droplet microfluidics record within hundreds to thousands of fps (Glawdel et al., 2012; Kemna et al., 2012; van Dijke et al., 2010; Wang et al., 2020). Consequently, each experiment which could be comprised of a few seconds worth of video, results in thousands of frames that require processing and analyzing. Commonly, this analysis is performed via traditional software, plugins and scripts run on platforms such as ImageJ or Matlab, and less frequently other highly-specific, proprietary analysis programs provided by microscope and camera manufacturers. Regardless of analysis approach, the most commonly extracted features from these datasets are the number of objects or droplets, size (area, diameter), shape (aspect ratio, circularity) and position. The basic extractable parameters of these droplets enable even further calculation of system dynamics such as velocity or localized concentration of droplets.

Droplet feature extraction is enabled by implementation of standardized image processing techniques such as the Circle Hough Transform for example (Illingworth & Kittler, 1987; Rizon et al., 2005; Yuen et al., 1990). The digital image data produced by observation of microfluidic droplets is usually ideal, as the droplet interface presents in high contrast against the chip background and continuous phase fluid. The large degree of control over experimental conditions enables the production of high-quality image data where focus and lighting are tuned so feature extraction is usually successful in providing highly representative system data (Vo et al., 2017; Zantow et al., 2013). As experimentalists begin to develop more complex microfluidic systems, the image-analysis portion of the experimental pipeline begins to bottleneck the process, similar to other disciplines which rely on feature extraction (Minervini et al., 2015). Complex chip architectures usually include either connections to external electrical components or on-chip features which induce specific droplet transport mechanisms. The addition of these chip features directly impede feature extraction in software suites such as ImageJ or Matlab. As experimental complexity scales, the image analysis portion of the pipeline not only increases computational and analysis time, but also requires the experimentalist to dedicate significant effort to data filtration and thresholding to enable accurate feature extraction. This manual pre-processing impacts the statistical robustness of emulsion or droplet data in more exotic microfluidic systems. The impact of this necessitates the development of new, smarter analysis methodologies that preserve statistical robustness, while limiting involvement in curation and treatment of raw data.

In order to facilitate a break-away from traditional, yet bulky image analysis pipelines, deep-learning methods have the potential to excavate the entrenched nature of these problems via the intrinsic handling and representation of raw data used in the slow, orthodox analysis. This untreated data can be fed to a model, which learns the necessary representations to tractably approach the task of classification or detection in a well-defined system of objects. The representations of the data are systematically passed through a series of modules, which perform transformations into subsequently more abstract representations (Goodfellow et al., 2016; LeCun et al., 2015). Within this network of abstract representation, complex functions emerge from these simple, yet non-linear modules which serve the purpose of enabling feature discrimination and suppression of irrelevant relationships in the decorrelated latent space (Dalal & Triggs, 2005; LeCun et al., 2015; Lienhart & Maydt, 2002). By feeding an image to this network, the latent space can begin to construct representations for the presence of certain pixels which ultimately contribute the formation of a feature structure. Subsequent abstraction can then take these suspected features and arrange ensemble combinations which correspond to the target objects, that are reconstructed through the generative portion of the network (Goodfellow et al., 2016; LeCun et al., 2015).

Since the emergence of deep neural networks (DNNs), there have been a number of machine- and deep-learning based tools developed for image analysis based on microscopy data, especially in the context of biological science. A prize application has been using these DNNs for time-intensive tasks such as cell detection and counting or even segmentation, and various model development has specifically tackled this (Berg et al., 2019; Van Valen et al., 2016; Waithe et al., 2020). Of these reports that used these DNNs for live cell imaging and eventual single-cell segmentation (Van Valen et al., 2016), the authors claim that this approach is not only more time efficient, but it also broke ground in regards to utilizing computer vision to analyze previously untenable co-culture experiments. Outside of singular instances where models were developed for specific tasks, entire analysis pipelines have been developed to handle end-to-end detection and analysis of objects common in microscopy such as cells (Belevich et al., 2016; Berg et al., 2019; Held et al., 2010; Luengo et al., 2017; Marée et al., 2016; McQuin et al., 2018; Paintdakhi et al., 2016; Suleymanova et al., 2018). For example, of these workflow packages, Suleymanova et al. (Suleymanova et al., 2018) developed an open-source software (FindMyCells) built on implementing DNNs to accurately detect astrocytes in immunohistological images. Once again, the common claim from the authors is that by benchmarking specific DNNs against traditional methods, there is an attainable outperformance by the former, where it almost achieves superhuman detection and classification capabilities.

In this work we leverage two implementations of anchor-based object detection CNNs, which are considered current state of the art models, that both approach the generative portion of the data pipeline differently. The original demonstrations of these two CNNs are: Faster R-CNN (Ren et al., 2015) and YOLO (Redmon et al., 2016). Below, we briefly discuss the models and show the generalized visualizations of the architectures in Fig. 1.

Faster R-CNN is one of the current, penultimate innovations on a series of CNNs (R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015)) which all approach the object detection problem via region proposal. Generally with this approach, regions of interest (RoI) are generated after a coarse scan of the input space followed by the computation of feature maps which are eventually passed to the softmax portion which proposes a classification for these RoIs. The Fast R-CNN architecture builds on the previous model by not only proposing RoIs, but generating a bounding box regressor in addition to the softmax class probability. However, region proposal in these CNNs is an expensive task which bottlenecks efficiency. Ren et al. (2015) then developed Faster R-CNN, one of the models used in this work, which grafts an additional region proposal network (RPN) to the backbone of the CNN which shares the convolutional feature maps with the detection portion of the model. This parallel RPN simultaneously communicates with the traditional CNN backbone while sliding over the transformed convolutional feature map which produces a low dimensional vector. This is passed to the fully connected layers which handle the two-pronged output of the box classification *cls* and the eventual transformation of the box regression layer *reg* to bounding box coordinates. As the RPN slides across the input space, each window has several proposals for regions of interest which eventually become the bounding boxes for the detected object, where the number of maximum proposals is *k*. The

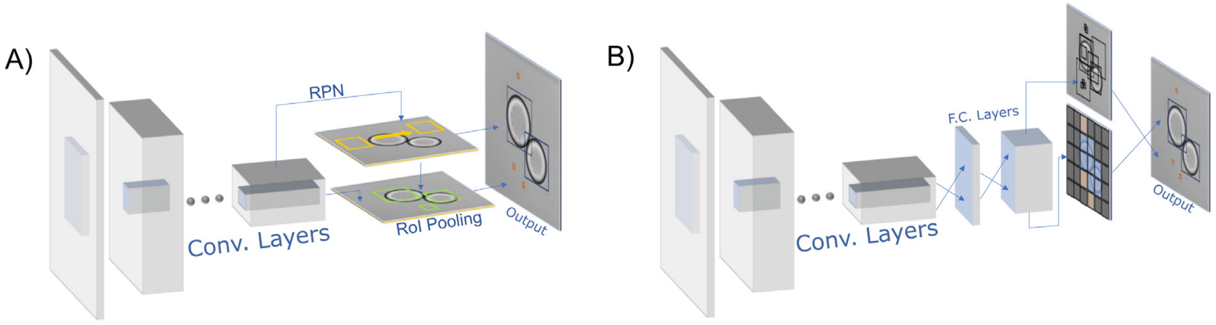

Fig. 1. Brief visual representations of the models used in this work. (A) Readapted graphical interpretation of the Faster R-CNN model. The first part of the model passes image input through a series of convolutional feature mapping layers. The backbone has a parallel region proposal network (RPN) grafted onto it which employs a sliding window that proposes regions of interest that are sent back into the pooling layer. The classification and bounding box regression loss proposals from the RPN are compounded to produce refined object classification and bounding box position tensors. (B) Similarly reduced graphical interpretation of the YOLO architecture. Images are also passed to a series of convolutional feature mapping layers before being passed to the fully connected (F.C.) layers. The unified inference output is a result of simultaneous performing of detection and classification. For a more comprehensive exploration of the model architectures, we encourage referencing the original literature which first proposed these models.

resulting outputs are via the aforementioned *reg* and *cls* layers, where the former has *4k* outputs generating the coordinates of *k* bounding boxes, and the latter handles *2k* estimates of probability of object (*obj*) or no object (*noobj*). The proposals are relative to the reference boxes, called anchors, which are centered at each window, yielding 9 total anchors at each window.

In contrast to interest region-based detection and classification, an alternative architectural "reapproach" to the object detection problem is framing it as a purely regression task. Various prototypical approaches (Erhan et al., 2014; Najibi et al., 2016; Pinheiro et al., 2015; Szegedy et al., 2014, 2013; Yoo et al., 2015), which tried to use this concept, attempted to address bottlenecks or speed issues with competing frameworks such as Faster R-CNN, but ultimately were hindered by limitations introduced in reducing the overall complexity of the architecture. Redmon et al. developed a model which eliminates the general repurposing of classifiers as detectors and handles the bounding box regression and class probability in a single backbone where you only look once (YOLO) at full image inputs (Redmon et al., 2016). The single throughput architecture is composed of 24 convolutional layers followed by 2 fully connected layers culminating in a $7 \times 7 \times 30$ tensor output of bounding box predictions. The reduction of complexity in YOLO enables the direct prediction of bounding boxes and associated class probabilities which in turn can take and train on full images and perform the task in a unified manner. Task unification is achieved by discretizing full images into parceled $S \times S$ grids which are then subsequently responsible for identification of potential object centroids within the corresponding grid cell. The bounding box predictions have an associated confidence which is defined as a product of probability Pr(Object) and IoU between the predicted box and the ground truth.

Biology and emulsion science naturally have overlap due to how both fields have been exploiting the advantages of the "lab-on-chip" aspect of microfluidics. There has been some research involving the integration of machine- and deep-learning methods into microfluidics which also investigate primarily cell imaging in flowing systems (Gavoille et al., 2019). As opposed to pipelining steady state microscopy for cell imaging via a DNN, (Heo et al., 2017) used a model which was able to classify flowing cells in real-time, opening the door for other integrations of DNNs in other lab-on-chip applications, such as flow properties. Hadikhani et al. (2019) were able to extend the application of a DNN to beyond biological applications, to measure fluid properties of droplets flowing in a microfluidic channel. They trained the network to accurately identify flow rate and mixture concentrations (water/isopropanol) of a dispersed phase based on the flow pattern of droplets. This enables extending these concepts to further exploit the efficiency of these DNN pipeline to generalized droplet detection and many more applications. Lastly, Zhang et al. have recently utilized Mask R-CNN architecture for detection of droplets in microchannels (Zhang et al., 2022). They compared the obtained size distributions with traditional methods and improved the overall analysis speed from ca. one hour to 3 s per image. Their study, however, was limited to only a single, relatively simple, channel structure. Based on these few, but very recent reports, we posit that in the coming years machine- and deep-learning will become a major area of focus for microfluidic researchers due to the immense potential for developing smart analysis and control pipelines that outperform traditional methods.

In this paper, we aim to demonstrate the viability of using various DNNs as tools for analyzing droplet based microfluidic data in the context of emulsion science. The detection of circular objects is a decades old computer vision task, inherently tied to microscopic observation, which has previously been universally approached by various implementations of the Hough transform or other computational solutions. As microfluidic architectures become more complex, the detection task also scales challenging traditional analysis methodologies by increasing the amount of resources necessary to shepherd data from beginning to end of pipeline. We show that the neural networks trained on custom microfluidic droplet data are robust and capable of universally handling the droplet detection task across various experimental systems and produce data which either rivals or outperforms traditional approaches, while significantly reducing user time involvement.

2. Methods

2.1. Microfluidic experiments

The data in this paper was obtained from analyzing a mix of microfluidic image data from previously published papers as well as newly produced data to benchmark the models against traditional methods. Therefore, the experimental methodologies used will only be discussed briefly and the reader is kindly referred to our prior published work for greater detail. The datasets for each specific type of analyzed phenomena were from the following sources: flow cells (Dudek, Bertheussen et al., 2018; Dudek, Chicault et al., 2020) and coalescence events (Dudek, Fernandes et al., 2020). The analysis performed for droplet generation, polydisperse flow cells, micrographs and porous media is unpublished. It should be noted that the same data from experiments (recordings) were analyzed by different methods. Therefore, all the labels in tables and figures refer to specific image analysis method used in the section, rather than different experiments.

For all microfluidic experiments with flow, the liquid flow rate was controlled with low- and mid-pressure pumps (Cetoni GmbH). The pumps were connected to a glass microfluidic chip, placed in a

Table 1
Selected experimental details for various type of microfluidic measurements presented in this work.

Experiment	Continuous phase		Dispersed phase		Frame rate [1/s]	Magnific./px-μm ratio
	Fluid	Flow rate [μl/min]	Fluid	Flow rate [μl/min]		
Drop generation	MQ water	50–250	Xylene with 0.1% wt. Span85	4	4000	10X/0.50
Flow cells (water)	3.5% wt. NaCl	160	Dodecane and crude oils	6–10	8500	10X/0.50
Flow cells (oil)	Crude oil	80	3.5% wt. NaCl	5	4000	10X/0.50
Flow cells (polydisperse)	1% wt. NaCl + 450 ppm Tween20	15	1-Br-dodecane	–	2000	15X/1.43
Porous media	1% wt. NaCl + 450 ppm Tween20	4	1-Br-dodecane	–	Single snapshots	10X/0.95
Coalescence events	3.5% wt. NaCl	160	Xylene with 0.1/1.0% wt. Span85	6	13 600	6X/0.30

microfluidic chip holder (all fabricated by Micronit Microtechnologies) via PEEK or PFA tubing and FFKM ferrules.

All experiments were recorded with a high-speed camera (AX100 or WX100, Photron), connected to an inverted microscope (Ti/Ti2-U, Nikon). The microscope is equiped with a motorized stage (OptiScan III, Prior Scientific, UK) that is coupled to the camera via LabVIEW code. Experiments involving micromodels (porous media) and micrographs were imaged by obtaining a sequential continuous grid of images using the motorized microscope stage. Some basic parameters of all experiments are displayed in Table 1.

Emulsions for the micromodel experiments were generated using a droplet generation chip comprising of a T-junction. A disperser was also utilized for producing the emulsion for the polydisperse flow cell experiment.

Two types of emulsions were used in the micrograph experiments: two monodisperse emulsions generated on the chip and two polydisperse emulsions prepared by means of the disperser. The monodisperse emulsions had droplet size of 45 μm and 23 μm magnification imaged at 10X (px-μm ratio = 0.95) and 15X (px-μm ratio = 1.43) magnification respectively. The polydisperse emulsions were referred to as coarse emulsion with mode droplet size of around 10 μm (10X magnification, px-μm ratio = 0.95) and fine emulsion with mode droplet size of around 2 μm (30X magnification, px-μm ratio = 2.85). The used dispersed and continuous phases are the same as for the experiments with porous media.

2.2. Reference image analysis

Drop generation and flow cells. After importing to ImageJ, stacks of images were thresholded and converted to binary. In some cases, contrast and brightness had do be adjusted manually prior to thresholding. Later, the drop sizes and positions were extracted with the "Analyze Particles" feature with a filter (size >10 μm and circularity >0.8). For flow cells, this data was then exported to Matlab, where a custom script produced size distributions and coalescence frequencies. This procedure is described in greater detail elsewhere (Dudek, Bertheussen et al., 2018; Dudek, Muijlwijk et al., 2018).

Coalescence events. Image sequences underwent a similar procedure as described above, however with a lower circularity filter (>0.4). In addition to areas, shape descriptors (aspect ratio, circularity, roundness) were additionally extracted. These were used to calculate the shape parameter, which allowed discrimination of standard, circular droplets from the coalescing objects. The procedure is described in greater detail in prior work (Dudek, Fernandes et al., 2020).

Micrographs. First, the micrographs were denoised using ImageJ's non-local means denoising plugin (smoothing factor = 2, sigma = 15) (Buades et al., 2011). Afterwards, the images underwent a three-step processing: (1) binarization by thresholding (dark background), (2) application of the "Fill Holes" feature in ImageJ to fill background pixels that are completely surrounded by foreground (droplet interface) pixels and (3) separation of touching droplets using the "Watershed" feature in ImageJ. The "Analyze Particles" function was used to obtain droplet number and sizes. The obtained data was processed and filtered in Matlab. The following filters were applied: (1) roundness <0.85, (2) circularity <0.8, (3) aspect ratio >1.2, and (4) diameter <5 pixels for polydispersed emulsions, and diameter <10 pixels for monodispersed emulsion.

Porous media. First, the images of the micromodel were binarized by thresholding (white background). Afterwards, "Analyze particles" was utilized to identify objects on the images, both droplets and pillars. The obtained data was filtered using a Matlab script: 38 <diameter <50 pixels for monodispersed; 15 <diameter <90 pixels for polydispersed. This filtering step allowed removal of data points representing pillars and satellite droplets from the dataset.

In cases when the Circular Hough Transform was used to detect droplets (flow cells [inlet], micrographs, porous media) the Matlab-native "imfindcircles" function was applied to the images using "Sensitivity" = 0.7-0.75 and "EdgeThreshold" = 0.1. For the experiments using micromodels the obtained data was filtered using Matlab script: 12 <diameter <50 pixels for monodispersed and 10 <diameter <85 pixels for polydispersed emulsions. Here, "EdgeThreshold" is a parameter that sets the gradient magnitude for determining edge pixels. The lower "EdgeThreshold", the more circular objects with weak edges will be detected. While "Sensitivity" is the main parameter which allows the function to identify weakly and partially obscured circular objects as circles. The higher the "Sensitivity", the more circular objects can be detected, however, higher "Sensitivity" can potentially lead to a greater number of false detections. The "EdgeThreshold" and "Sensitivity" values were identified empirically. The identified values are considered to be optimal for the used lighting conditions.

In one section (Micrographs), NIS-Elements BR software was used to extract droplet counts. Native functions (thresholding and object count), as well as filtering through limiting size and circularity, were used to extract the number of droplets in all images.

2.3. Neural network implementation

The CNNs described above are publicly available and three implementations were utilized in this work.

The Facebook AI Research group has maintained the repository which has been restructured under the Detectron2 umbrella which includes a model zoo for various CNNs including the Faster R-CNN version used in this work along with subsequent instance segmentation models such as Mask R-CNN (Wu et al., 2019). The entire Github repository is available at https://github.com/facebookresearch/detectron2. The specific version of Faster R-CNN implemented here is the "Faster R-CNN X-101-32x8d-FPN-3x" and for initializing the model, the pre-trained checkpoints for this same version were used to train our custom weights. The additional training parameters for this model were: 'image_batch_size = 4', 'base_learning_rate of 0.001', 'warmup_iterations = 1000', 'max_iterations = 4000', 'solver_steps = (1000, 1500)', 'gamma = 0.05'. The header parameters were: 'batch_size_per_image = 64' and 'num_classes = 2'.

Two latter versions of YOLO were used in this work compared to the original Darknet implementation by Redmon et al. (2016) available here: https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection. For the later versions of YOLO used in this work, we used YOLOv3 and YOLOv5 with more emphasis placed on the latter. YOLOv3 was mainly trained and deployed to compare against YOLOv5 which claims some of the weaknesses described earlier regarding identification of small and grouped objects were significantly improved upon. The YOLOv3 implementation was PyTorch based with training, validation and test data structured for the Keras API on the Darknet backbone (cloned and modified from: https://github.com/ultralytics/yolov3). To produce a fine-tuned set of weights for our custom microfluidic droplet data set, similar to Faster R-CNN, we initialized the model from pre-trained COCO weights available in the YOLOv3 repository. The following parameters were used to train YOLOv3: 'epochs = 500', 'batch_size = 16', 'img_size = [640, 640]', 'learning_rate = 0.001', momentum = 0.95', 'decay = 0.001' and 'num_classes = 2'. Number of classes across all datasets was 2, albeit the classes are different across datasets.

Our approach to training YOLOv5 on the custom droplet dataset was similar to YOLOv3, but the entire implementation, including dataset structuring, was PyTorch based. Our model was cloned and modified from the following repository: https://github.com/ultralytics/yolov5 (Glenn, 2020). The training was initialized with pre-trained COCO weights available through the YOLOv5 repository and the configuration file specified to be trained was specifically the YOLOv5s version of the model. The training parameters used for YOLOv5 were as follows: 'epochs = 1000', 'batch_size = 16', 'image_size = [416,416]', 'learning_rate = 0.001', momentum = 0.95', 'decay = 0.001' and 'num_classes = 2'. The main difference with our training approach to this newer version of YOLO is extending training to 1000 epochs due to how quickly the model trains as well as reducing image input size to 416 × 416 pixels. YOLOv5 is a significant achievement in terms of detection and overall inference speeds, as reports of >100 fps inference are common. We report equivalently quick per-frame inference (close to 150 fps) below in the results of our experimental investigation. With these types of inference rates, one can directly feed commensurate high-speed camera capture to the model which then produces on-the-fly data ready for analysis as the experiment is being performed.

2.4. Dataset construction

The final iteration of the generalized microfluidic droplet dataset used as the training and validation set at the time of writing was designed to represent a wide variety of droplets that could appear in a given two-phase microfluidic system. While being generally broad in terms of providing robust ground truth representations of droplets, we specifically excluded certain experiments which were then probed in our analysis where the droplet characteristics widely varied from the training set to investigate the inference breadth of the models. In other applied scenarios such as biological cell detection, the lighting conditions often heavily impact the detection metrics of a model (Van Valen et al., 2016; Waithe et al., 2020) and thus in terms of exposure we tried to maintain lighting consistency across our training set. Fig. 2 shows examples of images which were annotated and compiled into the training/validation/test set.

Shows an example of each type of image capture which was annotated and used in the general training/validation set. (A) Inlet Throat, (B) Porous Media, (C) Narrow Flow Cell (D) Monodisperse Flow Cell, (E) Polydisperse Flow Cell. (F) Also shown is a frequency distribution of droplets and a secondary "out_of_focus" droplet class in each of the image types in the generalized dataset.

A visual representation of how we annotated droplets from each microfluidic data subset is shown in Fig. 2. Unlike other computer vision applications whose datasets may include greater than 50 classes (Lin et al., 2014) for applications such as self-driving vehicles, our general dataset is strictly limited to droplets and the occasional appearance of an out-of-focus droplet in chips with larger channel depths. We perform the microfluidic experiments with tight control of the fluids being injected into the chips and thus we are mainly interested in droplets and thus the logic was to train the models to detect various varieties of this object. Out of focus droplets are not accurate representations of the valid droplets in our experiments, but usually present in visually similar ways and thus it was easy to include this as a separate class in our dataset to minimize the chance of the models falsely detecting inaccurate samples.

In the general dataset shown in Fig. 2, the biggest focus was to robustly represent the flow cell as it is the most common microfluidic tool to study various emulsified systems. In addition, we chose to represent a more complex detection task in the form of a porous media micromodel and the inlet throat. The chip architecture is well defined and presents a network of etch features that mimic the pixel ensembles which make up a droplet interface. The number of images included from each experiment is as follows: 'Inlet Throat = 485', 'Porous Media = 530', 'Narrow Flow Cell = 148', 'Monodisperse Flow Cell = 485' and 'Polydisperse Flow Cell = 255'. The base amount of total images in the generalized dataset is 1903 without augmentation. Out of 19,483 total annotated objects, out-of-focus droplets are very underrepresented in the generalized dataset at only 644 instances or 3.3%. By including this class in the dataset, our intention was to identify poorly captured droplets and easily remove them from the harvested droplet data.

The models were each trained on this dataset based on the parameters described in the previous section. The outcome of the training in regards to loss as a function of training iteration, for example, and other metrics such as mean average precision (mAP), recall are available in the supplementary information (Figures S1–S3). Confidence thresholds used during inference per experiment are also available in Table S1. After inference is performed, images are de-normalized by the native image resolution which allows extraction of bounding box coordinates and dimensions and transformation into micrometer values.

The workstation used for training and data analysis uses Windows 10, with 32 GB of DDR4 (3200 MHz) RAM (Corsair), Intel Core i9-10850K CPU running at 3.6 GHz and an Nvidia Geforce RTX 11 GB 2080Ti GPU.

3. Results and discussion

3.1. Droplet generation

Outside of the broader systemic applications of microfluidics, the intrinsic physical mechanisms, such as droplet generation, can be exploited via various chip architectures, where T-junctions and flow focusing geometries are the most common. The size and number of generated droplets depends on the flow characteristics of both phases, but also the properties of the fluid (Garstecki et al., 2006). Finding a system with appropriate drop size and number often requires several screening measurements, where various parameters are systematically tested (Fu et al., 2012; Husny & Cooper-White, 2006; Tice et al., 2003). This could entail, for example, a correct drop volume in droplet reactor systems or obtaining correct droplet size for emulsion generation. Many reports in the literature rely on specific drop size or monodispersity (Frenz et al., 2008; van Dijke et al., 2009), where the presented method would also be applicable. Here we show the results of a simple experiment: droplets were generated and recorded during a step-wise increase in the flow rate of the continuous phase through a T-junction (Fig. 3).

Droplet diameters obtained from ImageJ (solid line) and YOLOv5 (scattered points) for droplet generation experiment. The color of the points corresponds to their relative occurrence frequency in a given time unit. The line plot below shows the programmed changes in the flow rate of the continuous phase during the experiment.

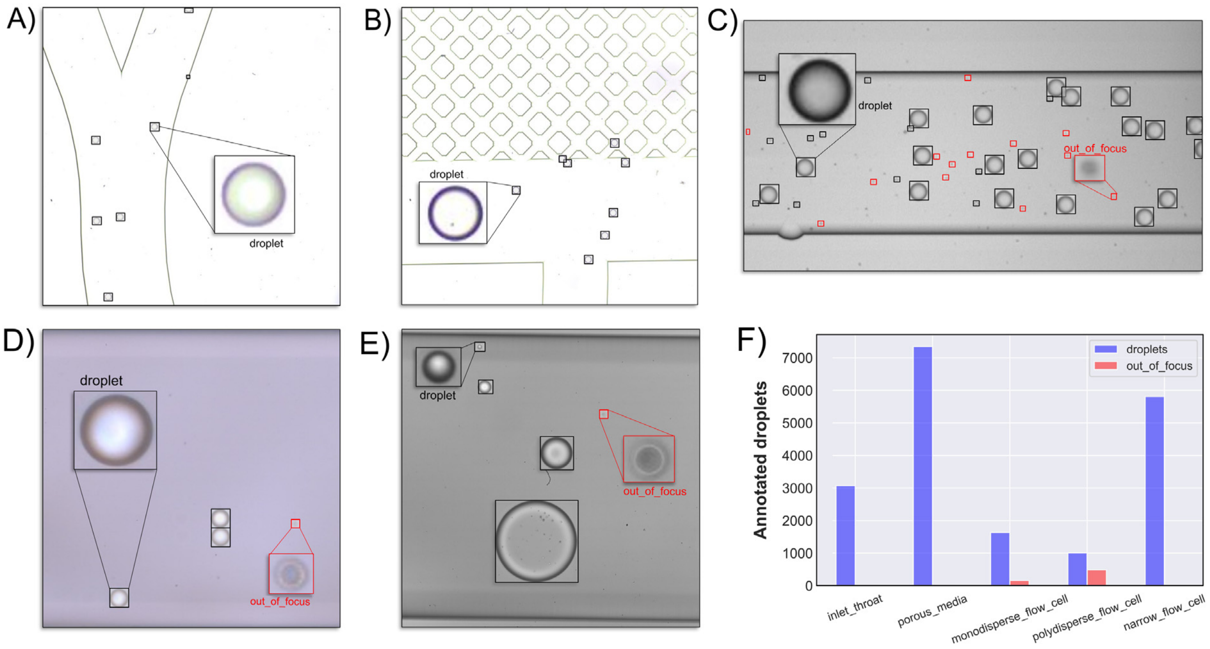

Fig. 2. Shows an example of each type of image capture which was annotated and used in the general training/validation set. (A) Inlet Throat, (B) Porous Media, (C) Narrow Flow Cell (D) Monodisperse Flow Cell, (E) Polydisperse Flow Cell. (F) Also shown is a frequency distribution of droplets and a secondary "out_of_focus" droplet class in each of the image types in the generalized dataset.

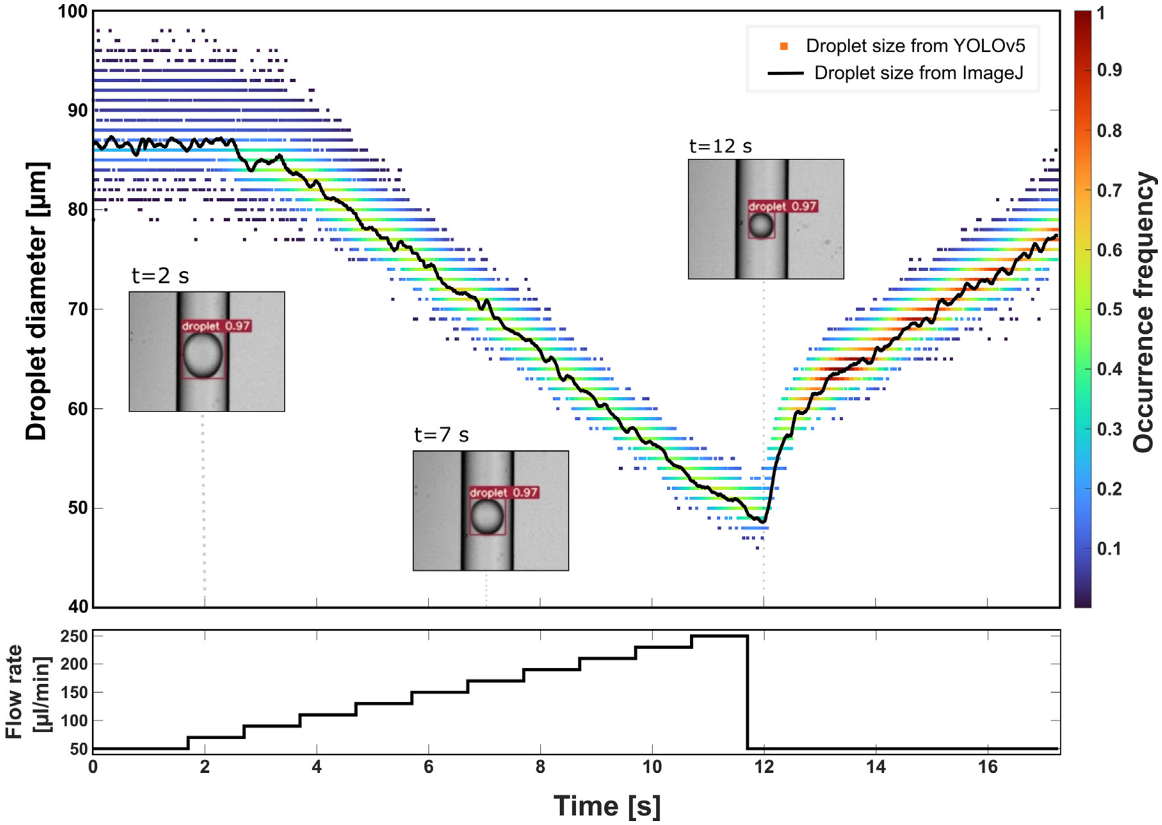

Fig. 3. Droplet diameters obtained from ImageJ (solid line) and YOLOv5 (scattered points) for droplet generation experiment. The color of the points corresponds to their relative occurrence frequency in a given time unit. The lineplot below shows the programmed changes in the flow rate of the continuous phase during the experiment.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

At time zero of the experiment, a constant continuous phase flow rate of 50 μl/min was introduced and maintained. After approximately 2 s, the continuous phase flow rate was ramped up by 20 μl/min per 1 s until reaching 250 μl/min, at which point the flow was dropped back to the initial level, without ramping. The scattered points on Fig. 3 are the values obtained via per-frame inference with YOLOv5. The color scheme is a heat-mapping which represents the relative occurrence of specific sizes in each time segment, i.e. dark and light blue-colored points signify diameters that did not appear often in the dataset, while orange- and red-colored points occurred very frequently. The black line shows the data obtained from complementary analysis in ImageJ. For the sake of image clarity, we used a moving average to smooth out this data.

The outcome of this droplet generation stress test raises several points. In the first two seconds of the measurement, the detected sizes ranged from 80 to 95 μm. However, the mode in this segment oscillates around the "ground-truth", reference values obtained from ImageJ measurements. Inspection of the annotated images showed that the bounding boxes were quite often not-precisely fit to the droplets. One possible reason for that was the proximity of droplets to the walls of the channel. In these flow conditions (shown in the first image inset of Fig. 3), the drops only had a few pixels worth of clearance in the horizontal direction. As flow rate is increased, expectedly the drop size decreases which also induced less fluctuation in the bounding box rendering. The precision deviation between droplet sizes corresponds to the proximity between the droplet interface and the edge features of the flow channel. The localized lighting conditions at the sub-5 μm scale could affect the model's ability to precisely discriminate between individual pixel ensembles belonging to the droplet or channel wall. A small lag in response between the change in the flow and the corresponding shift in drop size was observed, which could be attributed to the dead volumes in the tubing leading into the chip holder. At approximately 12 s of recording, the flow rate was brought back to the initial value of 50 μl/min. Consequently, the drop size started to increase, reaching approximately 80 μm by the end of the recording. Interestingly, the diameter distribution was observed to be the narrowest in the last segment of the measurement.

Despite the oscillatory nature of YOLOv5's detection precision, both the average detected diameter and the mode in each time segment shared symmetry to detection performed in ImageJ. The fast ramping of the flow rate, combined with dead volumes in the system resulted in a delayed response in the drop sizes, meaning that the drop size at each flow rate is probably not the "steady state" diameter that should be expected in these conditions. Nevertheless, these measurements show that one can obtain reliable, reasonably distributed size data via YOLOv5 inference. Lastly, one important limitation of ImageJ should be noted here. The entire recording comprised almost 70,000 individual frames. Loading and analysis of these amounts of images on standard PCs or laptops in ImageJ is essentially intractable. Even on our analysis PC (specifications described in Methods section), just loading all of the frames from memory into ImageJ consumes most of the RAM capabilities for more than 1 hr. In contrast, YOLOv5 took slightly longer than 10 min to entirely process and write detections for all 70,000 frames. Additionally, this lays the groundwork for various future applications where droplet-based systems in microfluidics are investigated in the context of using a CNN like YOLOv5, specifically capable of >100 fps inference (as in the experiment above). The model inference can be directly fed into a control algorithm which acts on the pumps to control flow for autonomous droplet or flow tailoring.

3.2. Flow cells

Subsequent to droplet generation in microfluidics, the various types of downstream droplet interaction are numerous and often the end goal of a microfluidic experiment, depending on the application. In most of our previous reports (Dudek, Bertheussen et al., 2018; Dudek, Ullaland et al., 2020), we focused on their coalescence or generally the interactions with other droplets. Here, we will showcase the analysis capabilities and symmetry to prior techniques using various CNNs on systems of freely flowing droplets, where the average diameter is smaller than the relatively large width (approximately 500 μm) of non-constrained flow cells. Our flow cell experiments can be divided into three subsets: (1) monodisperse droplets entering a wide channel shortly after generation (inlet of the coalescence chamber); (2) coalesced monodisperse droplets (outlet of the coalescence chamber); and (3) polydisperse droplets generated *ex-situ* and reinjected into a wide flow cell. Lastly, we will also present the detection of coalescence events during flow in microchannels. The droplets studied in each of these flow cell systems are shown in Fig. 4.

Table 2
The number of detected monodispersed droplets using various image analysis methods.

Experiment	Matlab	ImageJ	Faster R-CNN	YOLOv3	YOLOv5
OiW Model oil (weak contrast)	1627	1559	1605	1591	1595
OiW Crude oil (medium contrast)	2983	2911	2957	2636	2931
OiW Crude oil (strong contrast)	2805	2749	2797	2678	2788
WiO Crude oil (weak contrast)	1548	1223	1503	1475	1446

The first two (Fig. 4A and B) were typically part of one experiment, where the inlet and outlet of the coalescence channel was recorded in order to count the droplets coming in, and then count/measure the size of the droplets flowing out of the channel. One important aspect of these experiments was that the projected area of the droplets was increasing proportionally to the number of coalescence events that they underwent. This made it feasible to segregate droplets into size classes and calculate how many coalescence events occurred per time recorded. The data was then used to calculate the coalescence frequency, which is a common parameter to compare the stability of different emulsions (Dudek, Chicault et al., 2020; Krebs et al., 2012a, 2012b). In the contrasting case of polydisperse systems, the actual size distribution was of interest, as emulsions were prepared outside of the microfluidic chip and re-injected into the flow cell to record and later measure the droplet size.

Inference performed by YOLOv5 on (A) Monodispersed inlet, (B) Coalesced monodisperse (contrast), (C) Polydisperse flow cell, (D) Alternately trained YOLOv5 detections of teardrop and ellipsoidal coalescence events. Insets are simply magnified contents of the inference bounding boxes.

3.2.1. Monodisperse systems (inlet)

After generation, the droplets or bubbles are often transferred to wider channels, where they can interact with each other (Baret et al., 2009; Fu et al., 2015; Schröder et al., 2018; Wang et al., 2020). The initial part of these coalescence channels allows to measure their size and count (or estimate) the number of generated droplets. Table 2 summarizes the data obtained for monodisperse droplets at the inlet of our coalescence channel. Here we compare the three CNNs with two commonly used approaches for droplet analysis in microfluidics: feature extraction via Matlab and ImageJ. All methods were tested against four sets of fluid systems. Three of them were water-continuous with varying contrast of the dispersed oil phase, while the last one was oil continuous, meaning that the background (continuous phase) was considerably darker. It should also be noted that here all analyses were performed on every 150th image in the frame sequences recorded for the flow cells. Our previously reported drop counting method relies on calculating the drop velocity and later the average amount of appearances of a single drop while passing through an RoI box (Dudek, Muijlwijk et al., 2018). Since the recording speed of the videos is on the order of thousands of fps (depending on the total flow rate), the same droplet is spotted several times as it passes through the detection window, and for this reason the drop velocity was used to estimate the actual number of drops. However here, by re-analyzing sets of the filtered-out images (rather than whole sets), we could directly compare the detection efficiency of the CNNs and the overall analysis pipeline in contrast to the traditional methods.

Here we see that overall, both Faster R-CNN and YOLOv5 generally produce symmetrical analysis outcomes compared to ImageJ and

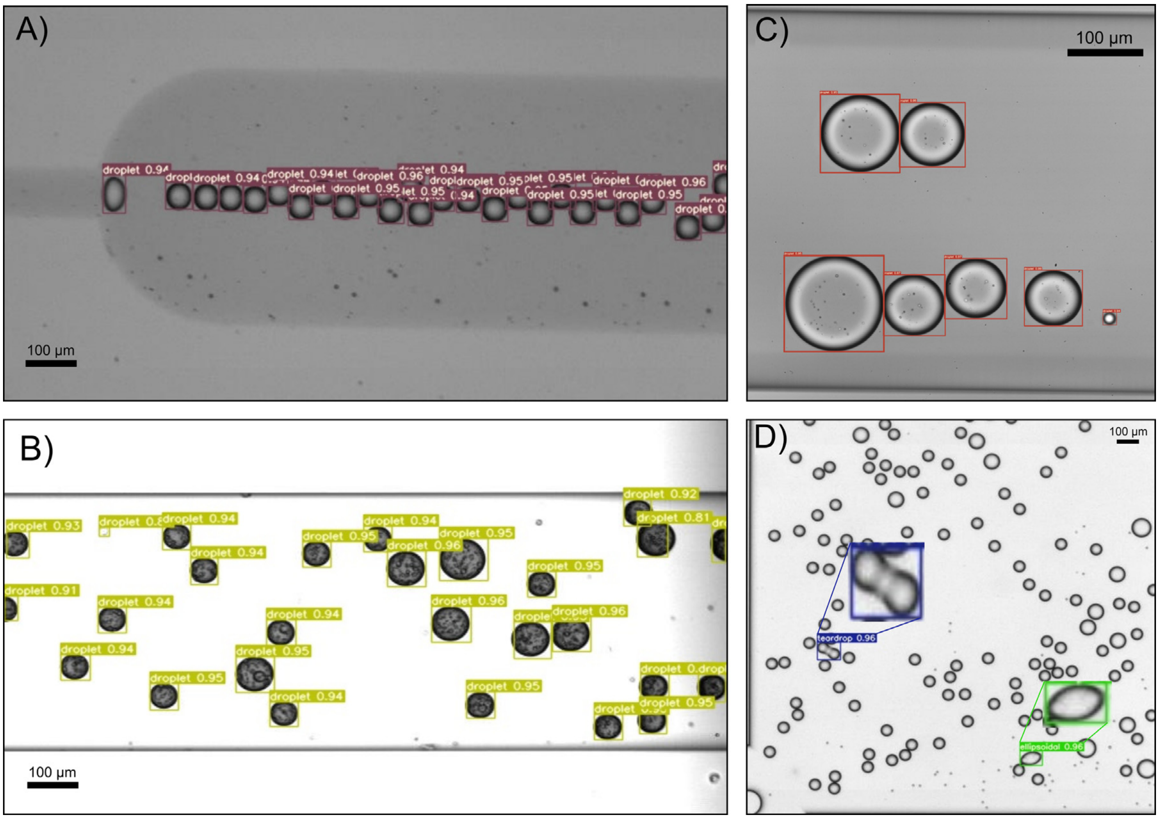

Fig. 4. Inference performed by YOLOv5 on (A) Monodisperse inlet, (B) Coalesced monodisperse (contrast), (C) Polydisperse flow cell, (D) Alternately trained YOLOv5 detections of teardrop and ellipsoidal coalescence events. Insets are simply magnified contents of the inference bounding boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Matlab. Based on visual inspection of the annotated images, detection with Matlab's implementation of the Hough transform seemed to be the most precise, and indeed this method always yielded the highest counts of drops. The values obtained from both Faster R-CNN and YOLOv5 closely followed those of the classic Hough transform, where in most cases the deviation was not higher than 1%. A similar system of a wider channel following drop generation junction, was also tested by Zhang et al. (2022), with even better drop detections, mostly due to some overlapping drops in the microchannels. We noted however, that YOLOv3 detected significantly less drops in 3 out of 4 tested sets. The inlet recordings typically contain trains of droplets in contact with one another, typically in one to three parallel rows, depending on the system and flow rates used (see Fig. 4A). When inspecting the inferenced images, visual verification showed that YOLOv3 failed to detect some droplets that were contained on the inner side of those droplet trains (as shown in Figure S4), consequently resulting in lower detection numbers. Interestingly, the oil-continuous system also yielded low droplet detections via ImageJ, compared to the complementary CNN analyses. Since the contrast between the oil and water phase is not as high as for the water-continuous systems, the thresholding of images in ImageJ was more demanding, and with small drop diameters, approximately 20% of objects were not detected. Additionally, the fit of the bounding boxes relative to the detected drops was significantly better when the channel was much wider than the drop diameter.

3.2.2. Monodisperse systems after coalescence (outlet)

After passing through the coalescence chamber, many of the initially monodisperse droplets can merge together and re-emerge as larger droplets later in the flow cell. Since the area of the droplets increase proportionally to the number of coalescence events, it is common to use size classes instead of actual droplet sizes (Dudek, Bertheussen et al., 2018; Krebs et al., 2012a, 2012b). The size classes are determined as follows: size class 1 is the initial droplet size, size class 2 is the droplet formed after coalescence of two size class 1 droplets, size class 3 is created by coalescence between size class 1 and 2 droplets, and so on. Fig. 5 shows the droplet size class distribution from one of the experiments, where image analysis was performed with all three CNNs and ImageJ as a reference. Subsequent systems with similar analysis are available in Figure S5.

Size distribution of initially monodisperse droplets at the outlet of the coalescence chamber obtained with all three CNNs and ImageJ for reference. Data for OiW Model oil (weak contrast).

As in the previous RoI velocity calculations, ImageJ was asked to perform analysis on full sets of experiment frames, while the three CNNs were fed every 150th frame. This gave some discrepancy between the absolute drop counts, since the image filtering factor was only an approximation based on the velocity of the droplets. Therefore, all size distributions were normalized for more transparent comparison. The actual final processing of the data into discretized size classes was performed in Matlab based on the inference from each model and ImageJ. All analysis methods provide similar results, with some small deviations for YOLOv3 at droplet size classes 4 and larger. Even though the boundaries for all size classes were equivalent, various singular instances of droplets were assigned to different size classes, across the CNNs. This was most likely the result of bounding boxes providing a non-precise fit 5 μm to the detected objects, which in turn affected their size, and consequently the calculated areas of the droplets.

While informative, drop size class distributions are not very effective to compare the extent of merging in various systems. In most of our reporting on this type of droplet system, we tend to use coalescence frequency as a single-value parameter for comparison. Briefly, coalescence frequency is calculated by dividing the ratio of the number of droplets at the inlet and outlet by the residence time. It describes how often, on average, a single droplet undergoes coalescence event per

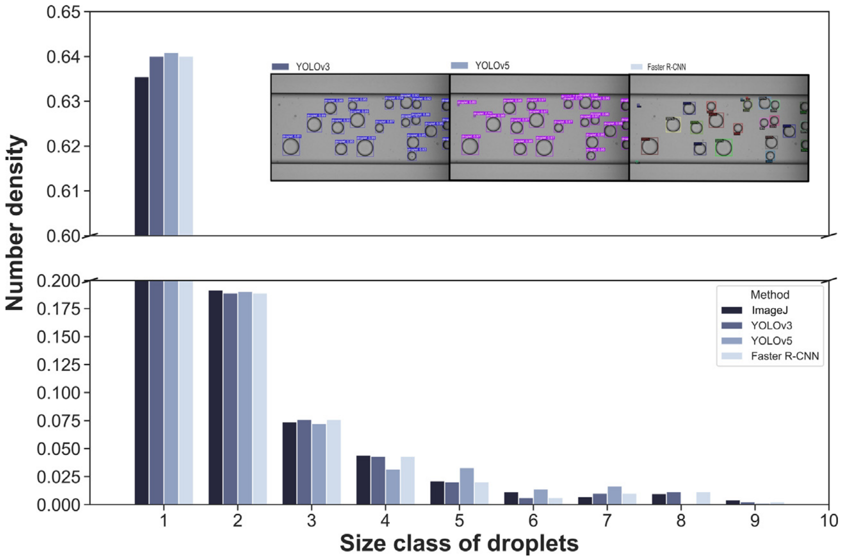

Fig. 5. Size distribution of initially monodisperse droplets at the outlet of the coalescence chamber obtained with all three CNNs and ImageJ for reference. Data for OiW Model oil (weak contrast).

Table 3
Coalescence frequencies calculated from the droplet size distributions using various methods.

Experiment	ImageJ	Faster R-CNN	YOLOv3	YOLOv5
OiW Model oil (weak contrast)	2.112	2.067	2.021	2.067
OiW Crude oil (medium contrast)	0.760	0.752	0.735	0.755
OiW Crude oil (strong contrast)	0.436	0.414	0.410	0.379
WiO Crude oil (weak contrast)	0.552	0.704	0.694	0.705

unit time. Hence, the higher the value of the coalescence frequency, the less stable emulsion is expected (i.e., more coalescence). Here, the analysis was also performed on selected systems with different merging characteristics (i.e., size distributions) and contrast between the continuous and dispersed phases (Table 3).

For this instance, we report three decimal points precision to probe if any significant deviation between analysis methods exists. In reality, the coalescence frequency values have to differ by ca. 20% minimum to observe significant differences between the systems, since the typical standard deviation between repeated measurements is typically not higher than 10%. Consequently, all three CNNs and ImageJ provided nearly equivalent coalescence frequency values for the first two cases (weak and medium contrast), where YOLOv3 was weaker as it provided slightly lower values. The other two cases are more interesting and therefore worth discussing. In the water-continuous system with strong contrast, image analysis via the CNNs yielded significantly lower results. The system in question is shown in Fig. 4B, with dark oil droplets presenting in a highly overexposed flow channel. Here, analysis with ImageJ was to some extent problematic, as it was difficult to properly threshold the images due to dark color of the oil and inner (water) droplets. This led to lack of detection of some initial-sized drops, which slightly increased the calculated coalescence frequency parameter. While Faster R-CNN and YOLOv3 seemed to detect all the droplets in the images for this system and gave a similar coalescence frequency, a number of the larger droplets were not detected by YOLOv5 (potentially as a result of high confidence thresholding). This caused the frequency parameter to be lowered by an additional 10% compared to the other two models. One possible reason for the lack of detection could be the inner features of the droplets, as for all of the other systems YOLOv5 was working equally, if not better than the other models. Both Faster R-CNN and YOLOv3 (especially the former) detected many more satellite droplets or image aberrations resulting from the flowing fluids or the microfluidic channel, leading to false detections. These detections can be forced by YOLOv5 if inference is again performed with lower confidence thresholding. However, for the sake of the analysis, satellites and false detections were filtered out of the analysis set. In the oil-continuous system, all the models provided very similar values, but higher than the result obtained from ImageJ. As in the previous flow cells, ImageJ struggles in certain systems where there is poor contrast provided between the droplets and the background. Therefore, we expect that the data harvested by the CNNs is in fact more accurate and reliable for sensitive phenomena in monodisperse flow cell systems.

3.2.3. *Polydisperse systems*

Monodispersed systems are highly controlled model experiments which offer insight into many fundamental droplet interactions but rarely approach addressing emulsified systems with a realistic size distributions. Imaging of less rigidly controlled emulsions in a microfluidic flow cell poses image processing challenges that are not characteristic in monodispersed droplets. We included an annotated set of polydisperse droplets in the overall training set, seen in Fig. 2, where various droplet morphologies and out-of-focus objects were prevalent. The inherent complexity of these polydisperse droplets increases especially in droplets with sizes greater than 100 pixels due to the high likelihood of droplets appearing within droplets—double emulsions. These very large droplets present at diameters sometimes 10 times greater than the smaller droplets. The goal of including this data was to teach the models that in polydisperse systems a droplet can exist at such varying levels of heterogeneous diameter, something that is unavailable in model monodisperse systems. At insufficient levels of training, precision as a function of training iterations has not converged enough with regards to being able to discern the interior droplets from the bulk droplet. However, as training iterations increase, with precise annotation, the models exhibit sufficient precision towards discerning double emulsions simply as whole droplets.

Additionally, when pivoting from narrow size distributions of droplets to emulsions with wide size distributions the focal plane becomes a factor which affects microscopic observation due to some of the droplets appearing out-of-focus (OOF). These droplets disappear fully or partially during the microscope observation of the flow and thus are omitted or inaccurately detected during image processing which contributes to statistical misrepresentation. Although, despite the models having the ability to identify out-of-focus droplets, the raw inference only visually distinguishes proposed bounds for the OOF droplet and thus a true size still cannot be ascertained. However, using DNNs for enhancement of images to super-resolutions is a highly researched area and could be a viable pathway for enhancing resolution for OOF droplets and investigating the frequency of their occurrence visually and the potential effect on the droplet statistics (Lim et al., 2017).

We performed a flow cell experiment where a polydisperse emulsion was generated off-chip and then re-injected in order to generate a high fps dataset to be fed to the traditional analysis tools as well as the models. In contrast to the monodisperse droplets where we fabricate size classes for relatively narrow distributions, the measured heterogeneous sizes can be represented as true size distributions. In this dataset we used pixel values to highlight the limits of each technique, regardless of the actual droplet size. The per-method inference visualization as well as the corresponding size distributions are shown in Fig. 6.

Droplet size distributions for a polydisperse emulsion observed in a flow cell. Each histogram shows the results of the detection and analysis for each method, (A) ImageJ, (B) Faster R-CNN, (C) YOLOv5, (D) YOLOv3. The inset of the upper secondary axis of each plot shows the results of the inference on the same frame from the experiment. (E) Detection frequency for each binned droplet size for ImageJ, Faster R-CNN and YOLOv5. YOLOv3 is not included due to insufficient detections of larger droplets.

The outcome of the analysis here shows excellent statistical symmetry, specifically, between processing performed in ImageJ and having YOLOv5 and Faster R-CNN infer the bounding boxes on the polydisperse droplets. ImageJ struggles with the smallest droplets that appear in this flow experiment while YOLOv5 and Faster R-CNN are able to identify an order of magnitude higher. These small detections make up the bulk of the sample sizes depicted in Fig. 6. YOLOv5 excels at detecting all small droplets <10 pixels and is still able to discriminate and classify non-usable OOF droplets. Faster R-CNN struggles with this which is why there are generally more droplets towards the small end of the total detections as it includes them into the results. YOLOv3 on the other hand is not a viable method, as it has up to this point struggled with very small object clusters and now also completely fails at detecting very large droplets and double emulsions. For this reason, the frequencies for YOLOv3 are not included in Fig. 6. The detections of this model are still acceptable for droplets within 35–200 pixels but outside of this range, YOLOv5 and Faster R-CNN outperform the older counterpart of the former.

Based on general histograms binned every 10 pixels, we wanted to examine the precision of the models at a more granular level of inspection between these sizes (mid-to-large droplets), without the skew from the small droplet sizes at the tail of the distributions. Fig. 6E shows the precision variation in terms of detection frequency as a function of droplet size for ImageJ, Faster R-CNN and YOLOv5. YOLOv3 is specifically omitted because of the poor detections of large droplets. Faster R-CNN is generally the model which reports more droplets from 100 pixels to approximately 400 pixels. All three methods converge well towards the extremely large sizes. The variation in frequencies between bins is best described by the spillover caused by imprecise bounding boxes. The same droplet which is detected on a per-frame basis in some cases could oscillate between the size bins as it travels in the flow channel based on the bounding box being rendered with imprecision between the box and the droplet interface. This precision oscillation when sequential frames are analyzed by the models is similar to what we observed in the droplet generation experiments in Fig. 3, which also relied on a very large dataset.

3.2.4. Coalescence events

In addition to just droplet detection, we show that the models can be trained to discriminate inter-droplet interaction morphologies, such as various coalescence events instead of the actual droplets themselves. Measurement of coalescence time is an important topic in the literature concerning emulsions, and several microfluidic techniques have been reported previously (Krebs et al., 2012a, 2012b; Wang et al., 2016, 2019; Zhou et al., 2016). In our previous work (Dudek, Fernandes et al., 2020), we have also reported a microfluidic method for measuring coalescence time. We define coalescence time as the time needed for the thin film, formed between two droplets in contact, to break. The coalescence time measurement was based on the detection of droplets in the process of coalescing, i.e. right after the moment of thin film breaking between the two drop interfaces, when the merged droplet returns to a typical, circular shape (see Fig. 4D). The detection of these droplet deformations was the starting point for our calculation of the coalescence time. With the help of several shape descriptors available through ImageJ particle analysis feature, it was possible to calculate a shape parameter for all the objects in the recording. As a result, out of approximately 50 000 frames and typically more than a million detected objects, one could filter out between 600 and 1000 coalescing droplets which were further analyzed. The image analysis performed in ImageJ is time consuming, due to the number of frames and size of the datasets, as previously highlighted.

Table 4
The number of detected coalescence events in different systems using ImageJ and YOLOv5.

Method	0.1% wt. Span85 @21 °C	1.0% wt. Span85 @21 °C	1.0% wt. Span85 @40 °C
ImageJ	886	725	878
YOLOv5	1068	904	1047
Same frame detections	798	600	382

Consequently, a new training set for YOLOv5 was prepared to enable a more precise and robust detection of coalescing droplets. We trained YOLOv5 on annotations which consisted of two morphologies of coalescence events: teardrop and ellipsoidal merging events. The *a priori* intention of developing weights for this detection task was to probe the ability of the models to discriminate between droplet sub-classes and not the actual droplets themselves. The training/validation/test (70/20/10% split) dataset for these coalescence events consisted of 786 images. The contributions from the each coalescence class consisted of 566 annotations for ellipsoidal events and 228 from teardrop shaped events. The model was trained with the same YOLOv5 parameters as the general droplet dataset described previously. An example of inference performed by YOLOv5 is shown in Fig. 4D, where an ellipsoidal and teardrop coalescence event occur on one frame. The number of detected coalescence events analyzed with YOLOv5 and ImageJ is presented in Table 4. Here, we show data for three different water-continuous systems, all with xylene droplets containing a nonionic surfactant (Span85) and at different temperatures (room or elevated temperature).

Depending on the parameters of the experiment, different numbers of coalescence events were detected. When increasing the surfactant concentration, the droplets were more stable and consequently less merging events were observed. Upon increasing the temperature, coalescence improved, which is to be expected as thermal treatment of emulsions is one of the common methods of inducing oil–water separation. In regards to the number of the detected coalescence events between the different methods, in all systems there was an approximate 20%–25% increase in events observed by YOLOv5. It should be noted that both datasets were processed in the same way as described in our previous work (Dudek, Fernandes et al., 2020), i.e. only events from frame 2 000 and above were included and the events from consecutive frames were removed. This data filtration had a larger effect on the results from YOLOv5, where often one coalescence event was detected over several frames due to deformation between droplets.

Table 4 also lists the number of same frame detections for both methods. Here, we compared the frame numbers associated with a detected coalescence event reported by both ImageJ and YOLOv5, and noted where the detections from both methods coincided. In the

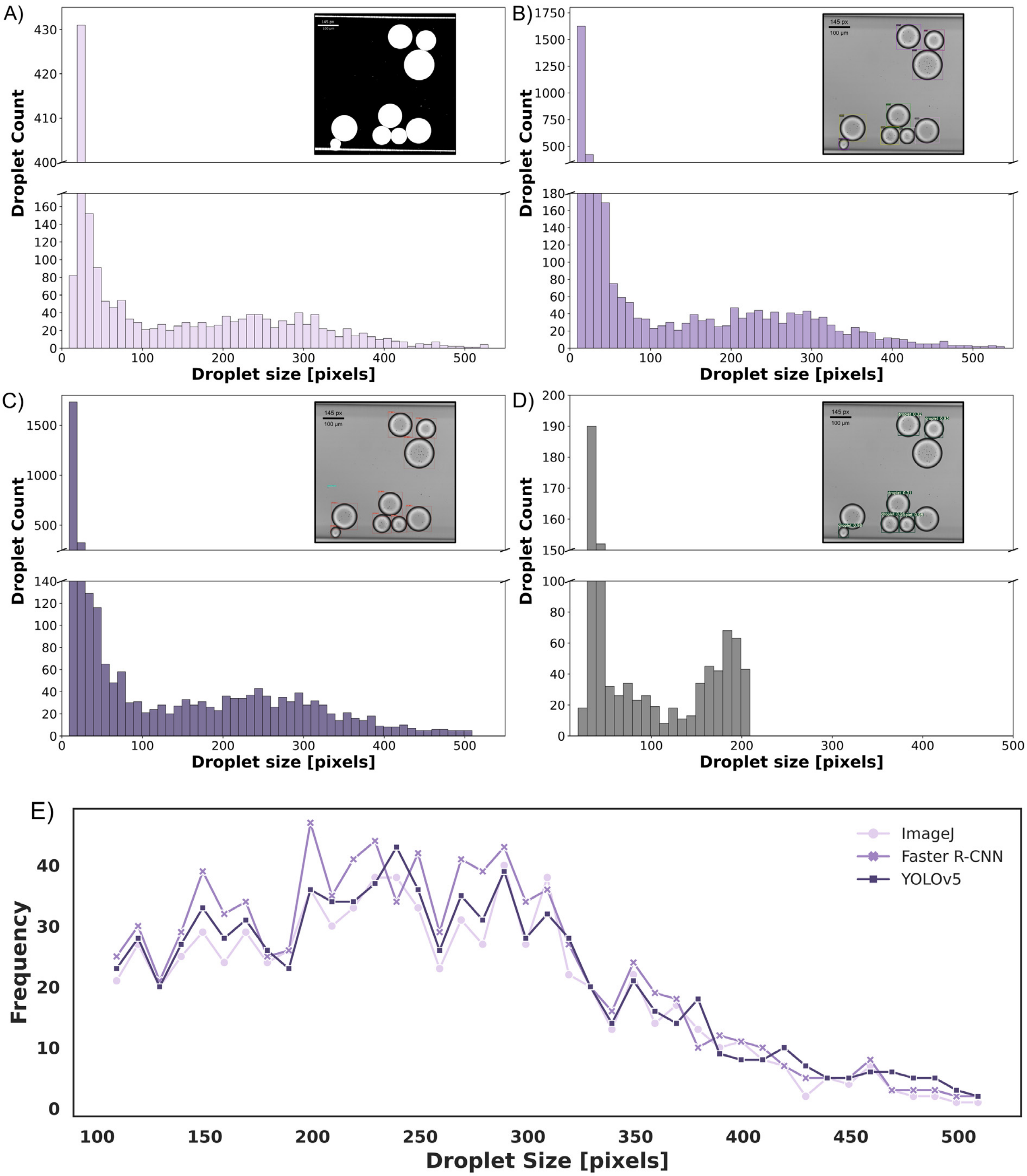

Fig. 6. Droplet size distributions for a polydisperse emulsion observed in a flow cell. Each histogram shows the results of the detection and analysis for each method, (A) ImageJ, (B) Faster R-CNN, (C) YOLOv5, (D) YOLOv3. The inset of the upper secondary axis of each plot shows the results of the inference on the same frame from the experiment. E) Detection frequency for each binned droplet size for ImageJ, Faster R-CNN, and YOLOv5. YOLOv3 is not included due to insufficient detections of larger droplets.

measurements at room temperature, these values are very comparable to the instances reported by ImageJ (ca. 100 lower for both systems), which means that YOLOv5 detected most of the same droplet merging events as ImageJ. The values for the higher temperature experiment are less symmetrical, however. One possible reason for that was that YOLOv5 was trained on two types of coalescence events (Fig. 4D), namely teardrop, where the droplets are caught in the moment of the film breakage; and ellipsoidal, where two drops are already in the process of merging and returning back to circular shape. Image analysis via ImageJ in most cases did not allow for detecting teardrop events, as after thresholding the boundary of the two coalescing droplets was discontinuous, and therefore omitted in the detection process. At higher temperature tests there seemed to be significantly more teardrop events detected with YOLOv5 (ca. 60% of total amount, compared to ca. 30% for the other two experiments at room temperature). Consequently, the same coalescence events could have been detected by ImageJ 2–3 frames after the detection with YOLOv5. And while the difference is quite small (few hundreds μs), this could also potentially lead to more accurate coalescence time measurements and eventually on-line measurements due to high inference speeds of YOLOv5.

Table 5
The number of detected monodispersed droplets using various methods.

Method	Matlab	ImageJ	NIS-Elements BR	Faster R-CNN	YOLOv3	YOLOv5
23 μm droplets	2638	2422	2375	700	1804	2100
45 μm droplets	1397	1223	1170	700	1360	1431

3.3. Micrographs

Thus far, the entirety of the droplet detection task has been framed in a microfluidics context. The models have been trained on purely microfluidic droplet data and then redeployed as analysis vehicles in other microfluidic systems. However, the dynamics regarding emulsion characteristics are often investigated via simpler avenues such as visual inspection after drop casting onto standard microscope slides (Colucci et al., 2020; Hu et al., 2017). Extracting data from these micrographs becomes a highly time-intensive process due to the image preprocessing and pipelining to commercial software or traditional analysis techniques, such as those discussed above. Hundreds of slices can be harvested from one drop-casted slide which can then yield greater than 50 000 detectable droplets, depending on the characteristics of the emulsion. Both approaches have drawbacks regarding pipeline efficiency and thus we posit that the acquisition speed of an automated stage coupled to the droplet detection models will not only allow for a significant increase in data acquisition rate but enable real-time acquisition. This is similar to a biological-focused implementation of Faster R-CNN and YOLO which used an automated stage that passed the microscope feed to the models which were trained to render detections of specific cells (Waithe et al., 2020). To implement this approach for emulsions on micrographs, single images can be simply passed to a persistent detection script or via batch processing. Once detection is invoked, of the three models used, Faster R-CNN has been shown to be the slowest while YOLOv3 and v5 inference speeds are less than 20 ms per frame. For YOLOv5, tuned with our weights, we reiterate that inference is typically 10 ms or less per frame on a GPU, and thus any potential bottlenecks will be related to the stage movement and subsequent image acquisition. With segmentation, however, the process can be significantly slower, as shown by Zhang et al. with their implementation of Mask R-CNN for droplet detection (Zhang et al., 2022).

Nonetheless, in terms of the actual droplet detection, we found that inference on micrographs was actually a more complex task than detection in the other microfluidic systems investigated. Conditions outside of a microfluidic system are highly variable and the residence of droplets within the liquid film on the microscope slide can contribute to varying levels of focus in the acquired image. Additionally, outside of a microfluidic chip there is a high tendency for micro-scale contaminants to enter the system at large. These factors contribute to natural image augmentation which is not present in a raw microfluidic training set. These augmentations coupled with the absence of annotated droplets coming from micrographs in the training set make detection more challenging for the models. Specifically, as in other CNN implementations for detection of microscopic objects, harsh lighting gradients often impact the inference capability of most models when fed raw images without prior image processing. This is evident as we benchmarked the CNNs against the traditional or commercial software on two monodisperse emulsions captured on micrographs with heavy vignetting. Table 5 shows the results of the total amount of droplets detected for each method and model. The vignetted micrographs for the 23 and 45 μm emulsions are available in supplementary Figure S6, which also shows the inferred droplet bounding boxes for each CNN.

The variability seen across these methods indicates that lighting conditions significantly affect droplet counting especially in the vignette fringes. Specifically, in the Matlab implementation of the circular Hough transform, despite the monodispersity of the emulsions, the shaded fringe droplets are detected but assigned an incorrect diameter which contributes to skewing of the known size distribution. In the 45 μm emulsion satellite droplets exist, which are also identified by the Hough transform and require filtering. YOLOv5 is the only of the models which also detects satellite droplets in addition to the intended droplets while YOLOv3 completely fails to identify the <10 pixel objects. Faster R-CNN is the weakest of the models as it only detects droplets in which it is only near 100% confidence without any thresholding applied during inference. Although, YOLOv5 had the most success in droplet identification and construction of a homogeneous size distribution, we also tested these natural micrograph augmentations in more heterogeneous droplet distributions, fine and coarse. We considered the first fine because the mode droplet diameter was approximately 2 μm. The second of the heterogeneous emulsions had mode diameter of approximately 10 μm which contributed to its coarseness. Fig. 7 shows the droplet size distributions for these emulsions characterized by heavily pre-processed images in ImageJ, as well as raw data inference performed by Faster R-CNN, YOLOv3 and YOLOv5.

Droplet size distributions shown by a probability density function for (A) fine emulsion with (B) YOLOv5 inference and (C) a coarse emulsion with (D) YOLOv5 inference. Inference results from ImageJ, YOLOv3 and Faster R-CNN are available in Figure S7.

Immediately, similarly to the microfluidic systems, there is statistical symmetry between the distributions produced by our traditional control method and mainly YOLOv5 in both emulsions. Despite the fine emulsion tending to be slightly more homogeneously dispersed, the dispersity is broad enough where there is variance between the software and the models when dealing with smaller droplets. In both emulsions, the distributions converge well at droplet diameters around 10 μm. However, the task is more challenging for the predecessor YOLOv3 and the region-based Faster R-CNN. Frequency counts suffer and the smallest droplets are completely omitted even at very low confidence thresholds (<50%) set during inference. In the case of finer droplets outside of microfluidic systems YOLOv5 is the superior choice for smaller, less familiar objects. As the droplet sizes grow along with a higher degree of polydispersity, the detection becomes similarly tractable for all three models where detection overlap is very high between ImageJ, YOLOv5 and Faster R-CNN. YOLOv3 still struggles due to the inherent architectural weaknesses which directly affects its inference capability for ultra-small objects (sub 10 pixels). The treatment of feature maps in YOLOv5 has been improved compared to older versions which still uses low resolution feature maps that contribute to YOLOv3's inability to detect ultra-small objects (Jocher, 2020).

The original intention with constructing a generalized droplet detection dataset was that it could broadly tune several CNN architectures for ubiquitous droplet detection across many systems. Clearly, as the detection task evolves to droplet systems outside the microfluidic realm, the natural augmentation and intrusion of disruptions to image quality suggest that the training task for both YOLOv5 and Faster R-CNN requires modification of the training data. Droplets are currently similar enough that the existing training set can tune the model to harvest droplet information with a competitive or better efficiency with significantly higher data acquisition rates (hours to seconds). Although we posit that higher degrees of precision can be achieved via inclusion of annotated micrographs in the model training or by assembling a training and validation set specifically for micrographs. Ultimately, the models struggle with droplets between 1 and 5 pixels, especially so in agglomerated groups, and thus it follows that model architecture must be addressed. Specifically, if there was some methodological unification between the discretized grid used by YOLOv5 and the sliding window of Faster R-CNN, higher resolution images can be fed to this hybrid architecture which moves across bigger images proposing detections within discretized grids. A non-architectural overhaul would simply involve experimental modification where higher microscope magnification is used with more sensitive stage movements, specifically for micrographs or porous media where precision loss is more prevalent in polydisperse systems, discussed below.

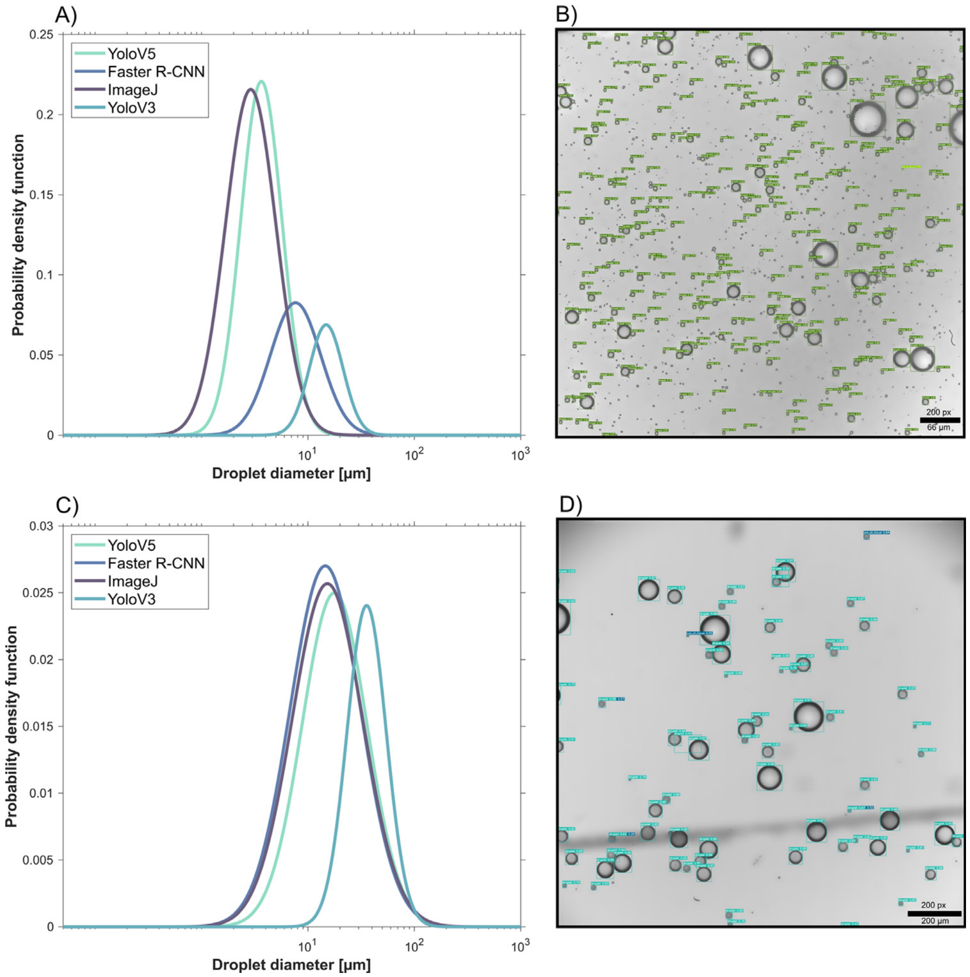

Fig. 7. Droplet size distributions for a polydisperse emulsion observed in a flow cell. Each histogram shows the results of the detection and analysis for each method, (A) ImageJ, (B) Faster R-CNN, (C) YOLOv5, (D) YOLOv3. The inset of the upper secondary axis of each plot shows the results of the inference on the same frame from the experiment. (E) Detection frequency for each binned droplet size for ImageJ, Faster R-CNN and YOLOv5. YOLOv3 is not included due to insufficient detections of larger droplets.

3.4. Porous media

Certain emulsion science phenomena often extend to understanding system dynamics outside of various flow regimes, such as in model systems with implications for extraction or retention characteristics (Auset & Keller, 2006; Liu et al., 2019). Microfluidics can be an alternative approach to conventional flooding techniques for studies concerned with the transport of emulsions in porous media (Azizov et al., 2022). Conducting these studies microfluidically allows pore-scale observations of droplet transport, whose outcomes could build a fundamental knowledge about droplet retention. However, such microfluidic experiments pose a challenge when it comes to image analysis as the microfluidic chips representing porous media have a complex geometry. The applicability of deep learning techniques in microfluidic porous structures has been previously shown in a recent report, where it was used to assess wettability changes in sandstone- and carbonate-type micromodels during flooding experiments (Yun et al., 2020). Here, we employ porous media studies in tandem with the integration of the CNNs into our image analysis pipeline for studying droplet flow in microfluidic porous media.

In order to simulate porous structures, the microfluidic chip architecture is modified into a topographical network of etched features intended to impede droplet transport, shown in Fig. 2B. The addition of the rock-like feature matrix raises the degree of complexity for traditional image analysis techniques. The complexity arises from the pixel intensities present at the edges of the etch features which are also highly similar to pixel ensembles present at droplets interfaces. The circular Hough transform can still be applied for droplet detection with a fair level of robustness, despite the presence of the porous matrix, albeit with stringent limitations. Firstly, as in the prior Hough transform implementations in our various microfluidic systems, the method struggles with detection of objects <5 pixels. Additionally, the etched features of the chip exhibited enough circularity to become challenging for the Hough transform to distinguish between them and droplets, and required sacrificing data integrity to filter them from the droplet statistics. The data processing is time and resource intensive, as typically processing time was 2.5–3 s per image, for a dataset of 25 000 images from various porous media experiments. ImageJ's "Analyze Particles" suite suffered from the same limitations as the Hough transform in these experiments and produced less quality data than the latter.

Despite the elevated level of background complexity present in these microfluidic systems, we posit the CNNs from the prior detection experiments can also be implemented as a competitive or superior method for droplet detection and data harvesting. Initially, in our prototype training and validation set, the models were not exposed to data from these porous media experiments. The best inference results had a broad mix of accurate detections of droplets and false detections of chip features. Similarly to the implementation of the Hough transform, where prominently convex chip features must be specifically filtered out, the detection of these edges must be accounted for by training out this type of false inference. The weights produced by our final training and validation dataset, shown in Fig. 2, droplets from various annotated porous media experiments were the most representative type of droplet

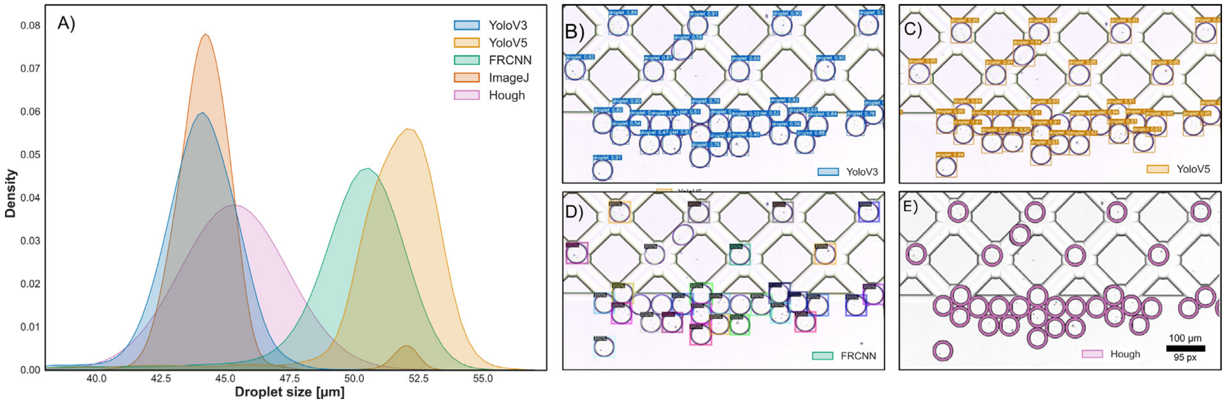

Fig. 8. (A) Shows kernel density estimation plot for all detection methods, including the traditional approaches, for a monodisperse emulsion injected into a porous media matrix. Inference results performed by (B) YOLOv3, (C) YOLOv5, (D) Faster R-CNN, (E) Hough Transform.

Table 6
The number of detected droplets in mono and polydisperse in porous media using various methods.

Emulsion homogeneity	Matlab	ImageJ	Faster R-CNN	YOLOv3	YOLOv5
Monodisperse	2989	2765	2658	2857	2844
Polydisperse	15 762	14 349	16 783	15 233	16 358

in the entire dataset. The nature of these experiments includes inhibition of droplet transport and thus the concentration of droplets at the face of the network is usually high compared to flow cell microfluidics, thus increasing the detection task complexity even further. Inference was performed on monodisperse droplets in porous media with all three CNNs, and the results are shown in Fig. 8. Absolute droplet counts can be seen in Table 6 for monodisperse and polydisperse droplets retained in the porous media chip, obtained for each detection method.

The glaring statistical difference between the reported monodisperse distributions is the appearance of two major diameter peaks. YOLOv3 and ImageJ both report that the majority of detected droplets have diameter centered at approximately 42.5 μm. The Hough transform shares slight overlap between these with the most Gaussian-like distribution which shared overlap between both mean peaks. Contrastingly, both Faster R-CNN and YOLOv5 mainly reported a bulk of the diameters approximately between 47 and 50 μm. The peak shift is explained by the placement of the detection or bounding box around or within the droplet. The Hough transform, ImageJ and YOLOv3 detect droplets by bounding within the droplet interface. The former two categorically eliminate the interface and thus shave several microns from a true estimation of the entire droplet. Faster R-CNN and YOLOv5 bound the droplet directly at the interface thus preserving the true diameter measurement without any further assumptive post-processing. Among these methods, despite the bounding precision of Faster R-CNN, it fails to detect significantly more droplets than all the other methods. We consider the Hough transform the reference method for droplet count or something very near "ground-truth" as we manually checked this entire dataset for missed detections. For this monodisperse set of droplets retained in the porous matrix, the Hough transform detected 2989 droplets. In terms of raw counts, YOLOv3 and YOLOv5 both outperform the entire ImageJ pipeline, which includes the "Analyze Particles" function and data post-processing. Generally, the Hough transform exists in the domain of accurate number of raw detections but algorithmically it performs detection of droplets within the interface, thereby shifting the true size distribution by approximately 8 μm, which is a drawback in ImageJ and YOLOv3 as well. Training YOLOv3 further will most likely result in some degree of overfitting for future predictions as precision and recall already converge well with loss over time, with the model reporting near 100% confidence for detected droplets. The overfitting risk exists within Faster R-CNN as well due to confidence levels of 100% despite a significant amount of missed detections. A solution could be to create a dataset specifically for porous media systems or other more complex microfluidic architectures that naturally augment the detection task with confusing edge morphologies that present droplet-like interfaces. However, YOLOv3 exhibits divergent behavior in the secondary porous media experiment, this time using a polydisperse emulsion. The results of this polydisperse retention experiment are shown in Fig. 9.

The polydispersity of the tested systems leads to a wide range of droplet diameters, and we thus use pixel values instead to emphasize the limitations of each detection method. Similarly to the monodisperse porous media results, we can use the Hough transform as a template for something near true droplet count in the system. Trend-wise even as the droplet size distribution broadens, there is distribution symmetry between YOLOv3 and ImageJ. The same methodological issue exists where for the Hough transform, ImageJ and YOLOv3 the droplets are mainly detected within their interfaces which again contributes to an underestimation of the true diameter on the order of 8 μm, approximately. It follows, that Faster R-CNN and YOLOv5 still generally infer the bounding box around the droplet interface. The level of polydispersity in this emulsion is clear due to multiple mean peaks which develop alongside bigger droplets beyond 50 μm diameter, which are mainly detected by Faster R-CNN and YOLOv5. YOLOv3 has the least "tail-like" behavior because it has the least consistent internal ruleset regarding bounding boxes as a function of droplet size. While it detects within the interface for larger droplets, as the diameter falls below 30 μm, YOLOv3 begins to overestimate the size of the droplet and fails at detecting small objects, as in the other experiments. Contrastingly, Faster R-CNN significantly overfits detections for smaller objects and often falsely detects either contaminants or small chip features with localized lighting artifacts as droplets contributing to a highly overestimated true droplet count. Most of this is observable in the tail under 10 μm. YOLOv5 also occasionally falsely reports artifacts or contaminants as small objects albeit with very low confidence compared to true small droplets with high confidence; therefore making it entirely feasible to filter false detections quickly. Even though we consider the Hough transform a relative baseline, it struggles with very small objects in this size range and we estimate that in more polydisperse droplet sets YOLOv5 is actually 5% more accurate at reporting true droplet counts due to the surprising detection aptitude across the entire size spectrum. ImageJ also struggles with small droplets for several reasons. Firstly, because the chip features for this microfluidic system must not be detected, ImageJ requires thresholding to do this and in the same pre-processing step, the smaller droplets are immediately removed from consideration. Additionally, the inherent convexity of the void space between circular droplets at agglomerated clusters is always detected and is also filtered.

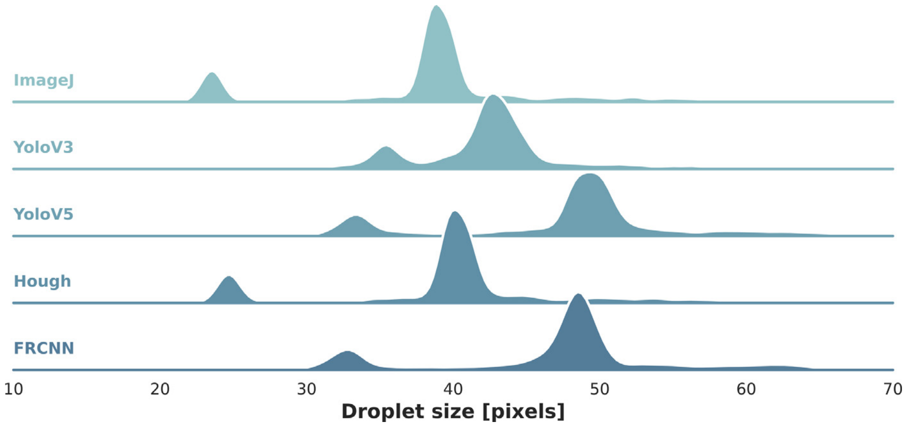

Fig. 9. Size distributions for data gathered from a porous media experiment which was injected with a polydisperse emulsion.

All methods falter when droplets appear in a microfluidic chip with features that are highly visually representative of droplet interfaces. Despite this, all methods generally report distribution symmetry, and being able to perform detection either inside or at the droplet interface is the main factor which affects the variance between the detections. Ultimately, both YOLOv3 and v5 can perform per-frame inference in less than 0.02 s which unlocks access to the same statistics as classical methods in fractions of the time required by the latter. A thorough assessment of finite time investment into post-experimental data treatment and processing with the classical methodologies is difficult to perform based on the variability, in terms of time required, by each individual to actually perform the task. More scrutiny can be placed on applying the correct filters to allow for a subjectively acceptable level of accurate object detection in ImageJ, while in the models, raw data is simply passed in without preprocessing and automatically utilized in experimental analysis.

4. Conclusions

Leveraging the computational advantage of various deep-learning approaches and CNN architectures in analysis pipelines is clearly evolving to be an intensely focused area of research. Throughout this work, we have shown that the three CNN's, Faster R-CNN, YOLOv3 and YOLOv5, can be trained on comprehensive microfluidic droplet datasets and then deployed as droplet detectors in a gamut of microfluidic systems and experiments. The models all show viability throughout the systems we tested, in comparison to data acquired via traditional methods and software packages i.e., Hough Transform implemented in Matlab and the Particle Analyzer suite in ImageJ). Overall, YOLOv5 appears more robust compared to the other two models, as it almost always provided precise bounding boxes in comparison to the traditional methods and outperformed these approaches as the detection task became more complex, as in the porous media systems. Faster R-CNN rivaled YOLOv5 in many of the experiments but sometimes overfit and falsely detected or completely missed obvious detections while also underperformed compared to YOLOv5's blazingly quick 10 ms per frame inference time. The former is still valuable as segmentation can be integrated via Mask R-CNN, and could potentially open the door for more complex levels of analysis in 3D droplet systems where the z-axis contributes to droplet agglomeration and occlusion. YOLOv5's consistent outperformance maintains statistical fidelity throughout the analysis pipeline at a fraction of the time investment and thus this method has become our primary droplet detection tool for many future microfluidic experiments. While the smallest version of YOLOv5 offers inference capable of live object detection at moderate framerates on high speed cameras, the model can be further condensed due to the fundamental nature of this well-established computer vision task. Decreasing the number of parameters will reduce the computational cost of pushing data through the model, thus decreasing its GPU memory footprint while still maintaining detection robustness in recording regimes greater than 200 fps.

CRediT authorship contribution statement

Gregory Philip Rutkowski: Conceptualization, Theoretical and experimental discussion, Experimentation and data collection, Software and script development, Formal data analysis, Writing – original draft, Writing – reviewing and editing. **Ilgar Azizov:** Conceptualization, Theoretical and experimental discussion, Experimentation and data collection, Traditional image analysis, Formal data analysis, Writing – original draft, Writing – reviewing and editing. **Evan Unmann:** Theoretical and experimental discussion, Software and script development, Writing – original draft. **Marcin Dudek:** Theoretical and experimental discussion, Experimentation and data collection, Traditional image analysis, Formal data analysis, Writing – original draft, Writing – reviewing and editing. **Brian Arthur Grimes:** Theoretical and experimental discussion, Writing – reviewing and editing, Supervision.

Data availability

The data used for this work including annotations for the droplet and coalescence training and validation sets is available upon request. The code used to train and analyze inference results is also available upon request.

Acknowledgments

The Research Council of Norway is acknowledged for the support to the Norwegian Micro- and Nano-Fabrication Facility, NorFab, project number 295864.

The authors would also like to acknowledge Gisle Øye for supporting the work and reviewing this manuscript.

Funding

The authors declare no competing financial interest.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.mlwa.2021.100222.

References

Anna, S. L. (2016). Droplets and bubbles in microfluidic devices. *Annual Review of Fluid Mechanics*, *48*(1), 285–309. http://dx.doi.org/10.1146/annurev-fluid-122414-034425.

Auset, M., & Keller, A. A. (2006). Pore-scale visualization of colloid straining and filtration in saturated porous media using micromodels. *Water Resources Research*, *42*(12), http://dx.doi.org/10.1029/2005WR004639.

Azizov, I., Dudek, M., & Øye, G. (2022). Studying droplet retention in porous media by novel microfluidic methods. *Chemical Engineering Science*, *248*, Article 117152. http://dx.doi.org/10.1016/j.ces.2021.117152.

Baret, J.-C., Kleinschmidt, F., El Harrak, A., & Griffiths, A. D. (2009). Kinetic aspects of emulsion stabilization by surfactants: A microfluidic analysis. *Langmuir*, *25*(11), 6088–6093. http://dx.doi.org/10.1021/la9000472.

Belevich, I., Joensuu, M., Kumar, D., Vihinen, H., & Jokitalo, E. (2016). Microscopy image browser: A platform for segmentation and analysis of multidimensional datasets. *PLoS Biology*, *14*(1), Article e1002340. http://dx.doi.org/10.1371/journal.pbio.1002340.

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., Eren, K., Cervantes, J. I., Xu, B., Beuttenmueller, F., Wolny, A., Zhang, C., Koethe, U., Hamprecht, F. A., & Kreshuk, A. (2019). Ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, *16*(12), 1226–1232. http://dx.doi.org/10.1038/s41592-019-0582-9.

Bremond, N., & Bibette, J. (2012). Exploring emulsion science with microfluidics. *Soft Matter*, *8*(41), 10549–10559. http://dx.doi.org/10.1039/C2SM25923K.

Buades, A., Coll, B., & Morel, J.-M. (2011). Non-local means denoising. *Image Processing on Line*, *1*, 208–212.

Colucci, G., Santamaria-Echart, A., Silva, S. C., Fernandes, I. P. M., Sipoli, C. C., & Barreiro, M. F. (2020). Development of water-in-oil emulsions as delivery vehicles and testing with a natural antimicrobial extract. *Molecules (Basel, Switzerland)*, *25*(9), 2105. http://dx.doi.org/10.3390/molecules25092105.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*.

Dudek, M., Bertheussen, A., Dumaire, T., & Øye, G. (2018). Microfluidic tools for studying coalescence of crude oil droplets in produced water. *Chemical Engineering Science*, *191*, 448–458. http://dx.doi.org/10.1016/j.ces.2018.07.006.

Dudek, M., Chicault, J., & Øye, G. (2020). Microfluidic investigation of crude oil droplet coalescence: Effect of oil/water composition and droplet aging. *Energy & Fuels*, *34*(5), 5110–5120. http://dx.doi.org/10.1021/acs.energyfuels.9b03434.

Dudek, M., Fernandes, D., Helno Herø, E., & Øye, G. (2020). Microfluidic method for determining drop-drop coalescence and contact times in flow. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, *586*, Article 124265. http://dx.doi.org/10.1016/j.colsurfa.2019.124265.

Dudek, M., Muijlwijk, K., Schroen, C. G. P. H., & Øye, G. (2018). The effect of dissolved gas on coalescence of oil drops studied with microfluidics. *Journal of Colloid and Interface Science*, *528*, 166–173. http://dx.doi.org/10.1016/j.jcis.2018.05.083.

Dudek, M., Ullaland, H. S., Wehrle, A., & Øye, G. (2020). Microfluidic testing of flocculants for produced water treatment: Comparison with other methodologies. *Water Research X*, *9*, Article 100073. http://dx.doi.org/10.1016/j.wroa.2020.100073.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *2014 IEEE conference on computer vision and pattern recognition*.

Frenz, L., El Harrak, A., Pauly, M., Bégin-Colin, S., Griffiths, A. D., & Baret, J.-C. (2008). Droplet-based microreactors for the synthesis of magnetic iron oxide nanoparticles. *Angewandte Chemie International Edition*, *47*(36), http://dx.doi.org/10.1002/anie.200801360.

Fu, T., Ma, Y., & Li, H. Z. (2015). Bubble coalescence in non-Newtonian fluids in a microfluidic expansion device. *Chemical Engineering and Processing: Process Intensification*, *97*, 38–44. http://dx.doi.org/10.1016/j.cep.2015.08.008.

Fu, T., Wu, Y., Ma, Y., & Li, H. Z. (2012). Droplet formation and breakup dynamics in microfluidic flow-focusing devices: From dripping to jetting. *Chemical Engineering Science*, *84*, 207–217. http://dx.doi.org/10.1016/j.ces.2012.08.039.

Garstecki, P., Fuerstman, M. J., Stone, H. A., & Whitesides, G. M. (2006). Formation of droplets and bubbles in a microfluidic T-junction—scaling and mechanism of break-up. *Lab on A Chip*, *6*(3), 437–446. http://dx.doi.org/10.1039/B510841A.

Gavoille, T., Pannacci, N., Bergeot, G., Marliere, C., & Marre, S. (2019). Microfluidic approaches for accessing thermophysical properties of fluid systems. *Reaction Chemistry & Engineering*, *4*(10), 1721–1739. http://dx.doi.org/10.1039/C9RE00130A.

Girshick, R. (2015). *Fast R-CNN proceedings of the 2015 IEEE international conference on computer vision (ICCV)*. http://dx.doi.org/10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE conference on computer vision and pattern recognition*.

Glawdel, T., Elbuken, C., & Ren, C. L. (2012). Droplet formation in microfluidic T-junction generators operating in the transitional regime. I. Experimental observations. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *85*(1 Pt 2), Article 016322. http://dx.doi.org/10.1103/PhysRevE.85.016322.

Glenn, J. (2020). Ultralytics/yolov5: v3.1 - Bug fixes and. http://dx.doi.org/10.5281/zenodo.4154370.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning (Vol. 1)*. MIT Press Cambridge.

Hadikhani, P., Borhani, N., H. Hashemi, S. M., & Psaltis, D. (2019). Learning from droplet flows in microfluidic channels using deep neural networks. *Scientific Reports*, *9*(1), 8114. http://dx.doi.org/10.1038/s41598-019-44556-x.

Held, M., Schmitz, M. H. A., Fischer, B., Walter, T., Neumann, B., Olma, M. H., Peter, M., Ellenberg, J., & Gerlich, D. W. (2010). CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature Methods*, *7*(9), 747–754. http://dx.doi.org/10.1038/nmeth.1486.

Heo, Y. J., Lee, D., Kang, J., Lee, K., & Chung, W. K. (2017). Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip. *Scientific Reports*, *7*(1), 11651. http://dx.doi.org/10.1038/s41598-017-11534-0.

Hiemenz, P. C., & Rajagopalan, R. (1997). *Principles of colloid and surface chemistry* (3rd ed.). CRC Press.

Hu, Y.-T., Ting, Y., Hu, J.-Y., & Hsieh, S.-C. (2017). Techniques and methods to study functional characteristics of emulsion systems. *Journal of Food and Drug Analysis*, *25*(1), 16–26. http://dx.doi.org/10.1016/j.jfda.2016.10.021.

Husny, J., & Cooper-White, J. (2006). The effect of elasticity on drop creation in T-shaped microchannels. *Journal of Non-Newtonian Fluid Mechanics*, *137*(1), 121–136. http://dx.doi.org/10.1016/j.jnnfm.2006.03.007.

Illingworth, J., & Kittler, J. (1987). The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(5), 690–698. http://dx.doi.org/10.1109/TPAMI.1987.4767964.

Jocher, G. (2020). *Ultralytics/yolov5: v3.1 - bug fixes and performance improvements*. Zenodo, https://github.com/ultralytics/yolov5.

Kemna, E. W. M., Schoeman, R. M., Wolbers, F., Vermes, I., Weitz, D. A., & van den Berg, A. (2012). High-yield cell ordering and deterministic cell-in-droplet encapsulation using Dean flow in a curved microchannel. *Lab on A Chip*, *12*(16), 2881–2887. http://dx.doi.org/10.1039/C2LC00013J.

Krebs, T., Schroën, K., & Boom, R. (2012a). A microfluidic method to study demulsification kinetics. *Lab on A Chip*, *12*(6), 1060–1070. http://dx.doi.org/10.1039/C2LC20930F.

Krebs, T., Schroën, K., & Boom, R. (2012b). Coalescence dynamics of surfactant-stabilized emulsions studied with microfluidics. *Soft Matter*, *8*(41), 10650–10657. http://dx.doi.org/10.1039/c2sm26122g.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://dx.doi.org/10.1038/nature14539.

Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *Proceedings. international conference on image processing*.

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*. Cham.

Liu, Q., Zhao, B., & Santamarina, J. C. (2019). Particle migration and clogging in porous media: A convergent flow microfluidics study. *Journal of Geophysical Research: Solid Earth*, *124*(9), 9495–9504. http://dx.doi.org/10.1029/2019JB017813.

Luengo, I., Darrow, M. C., Spink, M. C., Sun, Y., Dai, W., He, C. Y., Chiu, W., Pridmore, T., Ashton, A. W., Duke, E. M. H., Basham, M., & French, A. P. (2017). SuRVoS: Super-region volume segmentation workbench. *Journal of Structural Biology*, *198*(1), 43–53. http://dx.doi.org/10.1016/j.jsb.2017.02.007.

Marée, R., Rollus, L., Stévens, B., Hoyoux, R., Louppe, G., Vandaele, R., Begon, J. M., Kainz, P., Geurts, P., & Wehenkel, L. (2016). Collaborative analysis of multi-gigapixel imaging data using Cytomine. *Bioinformatics*, *32*(9), 1395–1401. http://dx.doi.org/10.1093/bioinformatics/btw013.

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B. A., Karhohs, K. W., Doan, M., Ding, L., Rafelski, S. M., Thirstrup, D., Wiegraebe, W., Singh, S., Becker, T., Caicedo, J. C., & Carpenter, A. E. (2018). CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biology*, *16*(7), Article e2005970. http://dx.doi.org/10.1371/journal.pbio.2005970.

Minervini, M., Scharr, H., & Tsaftaris, S. A. (2015). Image analysis: The new bottleneck in plant phenotyping [applications corner]. *IEEE Signal Processing Magazine*, *32*(4), 126–131. http://dx.doi.org/10.1109/MSP.2015.2405111.

Najibi, M., Rastegari, M., & Davis, L. S. (2016). G-CNN: An iterative grid based object detector. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*.

Paintdakhi, A., Parry, B., Campos, M., Irnov, I., Elf, J., Surovtsev, I., & Jacobs-Wagner, C. (2016). Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. *Molecular Microbiology*, *99*(4), 767–777. http://dx.doi.org/10.1111/mmi.13264.

Pinheiro, P., Collobert, R., & Dollar, P. (2015). Learning to segment object candidates.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

Rizon, M., Haniza, Y., Puteh, S., Yeon, A., Shakaff, M., Abdul Rahman, S., Sugisaka, M., Sazali, Y., Rozailan, M. M., & Karthigayan, M. (2005). Object detection using circular Hough transform.

Schröder, A., Sprakel, J., Schroën, K., Spaen, J. N., & Berton-Carabin, C. C. (2018). Coalescence stability of Pickering emulsions produced with lipid particles: A microfluidic study. *Journal of Food Engineering*, *234*, 63–72. http://dx.doi.org/10.1016/j.jfoodeng.2018.04.007.

Shang, L., Cheng, Y., & Zhao, Y. (2017). Emerging droplet microfluidics. *Chemical Reviews*, *117*(12), 7964–8040. http://dx.doi.org/10.1021/acs.chemrev.6b00848.

Suleymanova, I., Balassa, T., Tripathi, S., Molnar, C., Saarma, M., Sidorova, Y., & Horvath, P. (2018). A deep convolutional neural network approach for astrocyte detection. *Scientific Reports*, *8*(1), 12878. http://dx.doi.org/10.1038/s41598-018-31284-x.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection.

Tice, J. D., Song, H., Lyon, A. D., & Ismagilov, R. F. (2003). Formation of droplets and mixing in multiphase microfluidics at low values of the Reynolds and the capillary numbers. *Langmuir*, *19*(22), 9127–9133. http://dx.doi.org/10.1021/la030090w.

van Dijke, K., de Ruiter, R., Schroën, K., & Boom, R. (2010). The mechanism of droplet formation in microfluidic EDGE systems. *Soft Matter*, *6*(2), 321–330. http://dx.doi.org/10.1039/b916141d.

van Dijke, K., Veldhuis, G., Schroën, K., & Boom, R. (2009). Parallelized edge-based droplet generation (EDGE) devices. *Lab on A Chip*, *9*(19), 2824–2830. http://dx.doi.org/10.1039/B906098G.

Van Valen, D. A., Kudo, T., Lane, K. M., Macklin, D. N., Quach, N. T., DeFelice, M. M., Maayan, I., Tanouchi, Y., Ashley, E. A., & Covert, M. W. (2016). Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Computational Biology*, *12*(11), Article e1005177. http://dx.doi.org/10.1371/journal.pcbi.1005177.

Versluis, M. (2013). High-speed imaging in fluids. *Experiments in Fluids*, *54*, 1458. http://dx.doi.org/10.1007/s00348-013-1458-x.

Vo, P. Q. N., Husser, M. C., Ahmadi, F., Sinha, H., & Shih, S. C. C. (2017). Image-based feedback and analysis system for digital microfluidics. *Lab on A Chip*, *17*(20), 3437–3446. http://dx.doi.org/10.1039/C7LC00826K.

Waithe, D., Brown, J. M., Reglinski, K., Diez-Sevilla, I., Roberts, D., & Eggeling, C. (2020). Object detection networks and augmented reality for cellular detection in fluorescence microscopy. *Journal of Cell Biology*, *219*(10), Article e201903166. http://dx.doi.org/10.1083/jcb.201903166.

Wang, T., Andersen, S. I., & Shapiro, A. (2020). Coalescence of oil droplets in microchannels under brine flow. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, Article 124864. http://dx.doi.org/10.1016/j.colsurfa.2020.124864.

Wang, J., Tan, S. H., Nguyen, A. V., Evans, G. M., & Nguyen, N.-T. (2016). A microfluidic method for investigating ion-specific bubble coalescence in salt solutions. *Langmuir*, *32*(44), 11520–11524. http://dx.doi.org/10.1021/acs.langmuir.6b03266.

Wang, J., Teo, A. J. T., Tan, S. H., Evans, G. M., Nguyen, N.-T., & Nguyen, A. V. (2019). Influence of interfacial gas enrichment on controlled coalescence of oil droplets in water in microfluidics. *Langmuir*, *35*(10), 3615–3623. http://dx.doi.org/10.1021/acs.langmuir.8b03486.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Yoo, D., Park, S., Lee, J.-Y., Paek, A. S., & So Kweon, I. (2015). Attentionnet: Aggregating weak directions for accurate object detection. In *Proceedings of the IEEE international conference on computer vision*.

Yuen, H. K., Princen, J., Illingworth, J., & Kittler, J. (1990). Comparative study of Hough Transform methods for circle finding. *Image and Vision Computing*, *8*(1), 71–77. http://dx.doi.org/10.1016/0262-8856(90)90059-E.

Yun, W., Liu, Y., & Kovscek, A. R. (2020). Deep learning for automated characterization of pore-scale wettability. *Advances in Water Resources*, *144*, Article 103708. http://dx.doi.org/10.1016/j.advwatres.2020.103708.

Zantow, M., Dendere, R., & Douglas, T. S. (2013). Image-based analysis of droplets in microfluidics. In *Annu int conf IEEE eng med biol soc, Vol. 2013* (pp. 1776–1779). http://dx.doi.org/10.1109/embc.2013.6609865.

Zhang, S., Liang, X., Huang, X., Wang, K., & Qiu, T. (2022). Precise and fast microdroplet size distribution measurement using deep learning. *Chemical Engineering Science*, *247*, Article 116926. http://dx.doi.org/10.1016/j.ces.2021.116926.

Zhou, Q., Sun, Y., Yi, S., Wang, K., & Luo, G. (2016). Investigation of droplet coalescence in nanoparticle suspensions by a microfluidic collision experiment. *Soft Matter*, *12*(6), 1674–1682. http://dx.doi.org/10.1039/C5SM02924D.