Machine Learning-Enabled Multiplexed Microfluidic Sensors

Sajjad Rahmani Dabbagh[1,2], Fazle Rabbi[1], Zafer Doğan[3,4], Ali Kemal Yetisen[5], and S. Tasoglu[1,2,6,7,¥]

[1]Department of Mechanical Engineering, Koç University, Sariyer, Istanbul, Turkey 34450.

[2]Koç University Arçelik Research Center for Creative Industries (KUAR), Koç University, Sariyer, Istanbul, Turkey 34450.

[3]Department of Electrical and Electronics Engineering, Koç University, Sariyer, Istanbul, Turkey 34450.

[4]Koc University Is Bank Artificial Intelligence Lab (KUIS AILab), Koç University, Sariyer, Istanbul, Turkey 34450.

[5]Department of Chemical Engineering, Imperial College London, London SW7 2AZ, UK

[6]Boğaziçi Institute of Biomedical Engineering, Boğaziçi University, Çengelköy, Istanbul, Turkey 34684.

[7]Koc University Research Center for Translational Medicine, Koç University, Sariyer, Istanbul, Turkey 34450.

[¥]Corresponding Author: Savas Tasoglu (stasoglu@ku.edu.tr)

Abstract

High-throughput, cost-effective, and portable devices can enhance the performance of point-of-care tests. Such devices are able to acquire images from samples at a high rate in combination with microfluidic chips in point-of-care applications. However, interpreting and analyzing the large amount of acquired data is not only a labor-intensive and time-consuming process, but also prone to the bias of the user and low accuracy. Integrating machine learning (ML) with the image acquisition capability of smartphones as well as increasing computing power could address the need for high-throughput, accurate, and automatized detection, data processing, and quantification of results. Here, ML-supported diagnostic technologies are presented. These technologies include quantification of colorimetric tests, classification of biological samples (cells and sperms), soft sensors, assay type detection, and recognition of the fluid properties. Challenges regarding the implementation of ML methods, including the required number of data points, image acquisition prerequisites, and execution of data-limited experiments are also discussed.

Keywords: machine learning; microfluidics; multiplexed sensors; diagnostics; point-of-care testing

1. Machine Learning and Implementation Prerequisites

1.1. Machine learning

Machine Learning (ML), as a subdivision of Artificial Intelligence (AI), enables computers to learn using example data or past experiences without being explicitly programmed. ML uses the theory of statistics in building mathematical models defined up to some parameters, and the learning process is executed by a computer program to optimize the parameters of the model based on classes of tasks and performance measures (Figure 1) [1-3]. One way to categorize ML algorithms is based on how they interact with the example data or experience. Accordingly, ML algorithms can be grouped by the learning style as supervised, unsupervised, and reinforcement learning. In supervised learning, the training data can be defined as an input X and an output Y (with a known/desired label or result), and the task is to learn the mapping from the input to the output. Regression and classification problems are examples of supervised learning problems. The training process continues until the model achieves a desired level of accuracy on the training data. Unsupervised learning does not employ labeled or supervised output data. In this case, the aim is to find structures and regularities in the input data. Clustering, segmentation, dimensionality reduction, and association rule learning are examples of unsupervised learning approaches. Figure 2 depicts the commonly used ML architectures. Semi-supervised learning methods are also used when the input data is a mixture of labeled and unlabeled examples. However, reinforcement learning algorithms do not require labeled or unlabeled input/output pairs, but focus on optimizing an output policy – defined as a mapping from state actions that provides instructions in a given state with a sequence of states and actions with (delayed) rewards [4, 5]. Although the supervised learning method is an accurate, effective, and versatile approach, the main drawback of this method is its reliance on prior knowledge - prepared labeled data by a human to train the algorithm, which is susceptible to human bias and demands substantial time and effort for preparation [6]. In some cases, the needed training input data for supervised ML methods can be obtained from the output of unsupervised ML approaches [7].
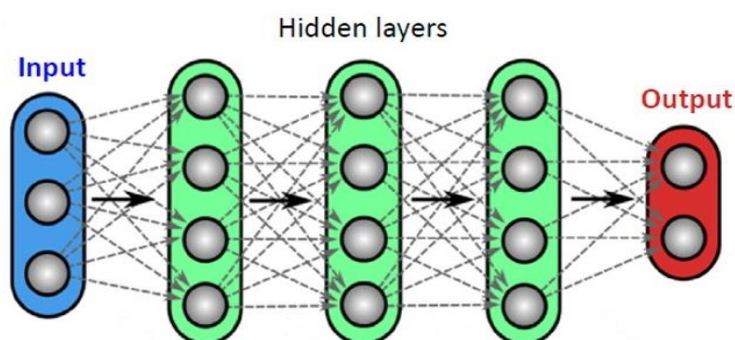

Figure 1. A schematic of a fully connected neural network (NN) comprised of input, output, and hidden layers. In conventional ML algorithms, data is first represented in terms of specific features that will allow for dimensionality reduction. However, based on the underlying mathematical model, current Deep Learning (DL) models can be trained without necessarily hand-picking such features. In NNs, as the number of hidden layers increases, the network becomes deeper [8]. Shallow NNs (with few numbers of hidden layers) have limited modeling capability that is suitable for simple and well-structured data. However, multilayer, deep algorithms possess the required complexity for undertaking more real-life tasks. Reproduced with permission from J. Riordon, D. Sovilj, S. Sanner, D. Sinton and E. W. Young, Trends Biotechnol. 37 (3), 310-324 (2019). Copyright 2018 Elsevier Ltd.
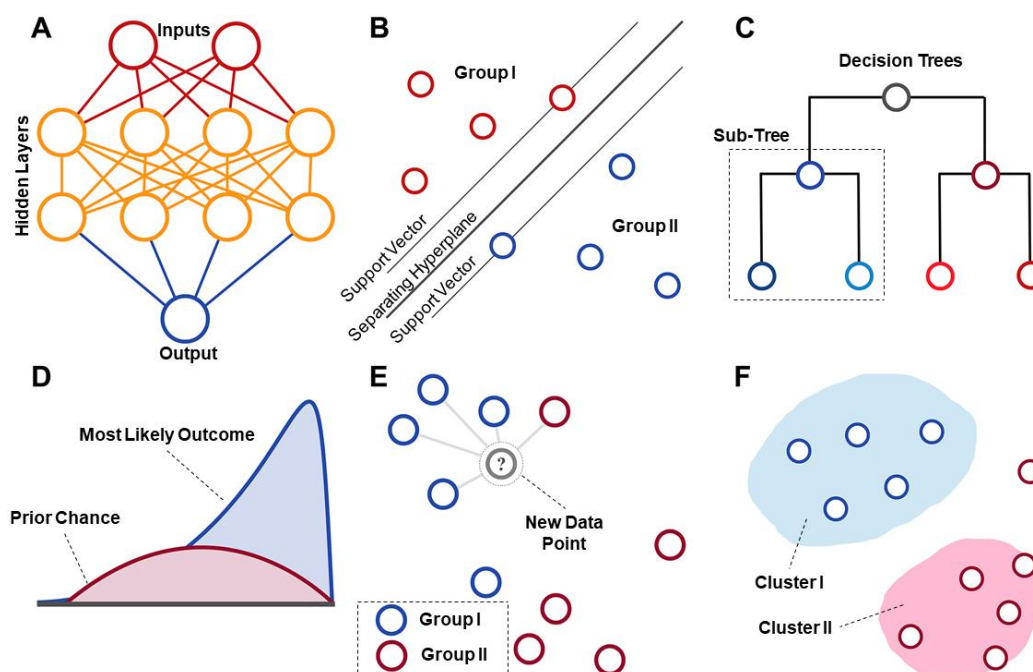
Figure 2. Most common ML algorithms. (A) ANNs are inspired by the NNs in the brain, structured in layers of interconnected nodes. The nodes in the red layer are the input features, nodes in the orange layers are hidden layers, and the node in the blue layer is the distinct output. Although ANNs can model complex relationships between in- and output features, the interpretation of how an algorithm reaches the output from an input is still difficult. (B) Support vector machine (SVM) is a supervised ML algorithm that uses the classification of data points by choosing the "separating hyperplane" that maximizes the distance from the 2 closest points on either side to increase the generalizability to unseen data. (C) Decision trees use bifurcating of the feature space to make classifications or predictions based on numerous input features. While regression trees are a result of continuous decision variables, the categorical decision variable produces classification trees. In order to solve the overfitting of a single tree, random forests, as an ensemble learning method, takes the mean predictions of the individual trees or the mode of classes. (D) Naïve Bayes calculates the most likely outcome (blue) as a product of the a priori chance (red) and the conditional probabilities given by the individual features, which is usually not definitely true, but generally is rapidly computed and provides viable prediction in practice. (E) A data point, with an unknown class, is compared to its K nearest neighbors by the K-Nearest Neighbors in order to determine its class as the most common class of its neighbors. For K = 1, the algorithm assigns the class of a data point to the class of the single closest neighbor. (F) Fuzzy C-Means, as an unsupervised learning algorithm, can cluster data points without having the desired output, based on their input features. Owing to the "fuzzy" aspect, these algorithms are flexible to classify a data point to each cluster to a certain degree relating to the possibility of fitting to that cluster [9]. Reproduced with permission from J. T. Senders, P. C. Staples, A. V. Karhade, M. M. Zaki, W. B. Gormley, M. L. D. Broekman, T. R. Smith and O. Arnaout, World Neurosurg. 109, 476-486.e471 (2018). Copyright 2017 Elsevier Inc.

1.2. Deep learning strategies

DL is a subset of a broader family of ML methods based on Artificial Neural Networks (ANNs). As such, learning can be supervised, semi-supervised, or unsupervised. For supervised learning tasks, deep learning methods eliminate the need to design or select good features that are domain-specific, by translating the data into compact intermediate representations similar to principal components, and derive layered structures that remove redundancy in representation. While supervised DL models need a large amount of labeled training data to enhance their accuracy, they demonstrate superior performance as well as a significant practical benefit with respect to conventional ML methods in several data-driven applications such as image classification,

segmentation, object detection, face recognition,. revealing the power of having access to more data. DL algorithms can be applied to unsupervised learning tasks as well. This is an important benefit because unlabeled data are more abundant than the labeled data. Moreover, these algorithms derive insights directly from the data itself, by summarizing and grouping the data, so that one can use these insights to make data-driven decisions. Several unsupervised DL models exist, such as Autoencoders, Deep Belief Nets, Hebbian Learning, Generative Adversarial Networks (GANs), and Self-organizing maps that do not rely on labeled training data to make decisions and determining the accuracy of the outcome. Despite demanding considerable time for training in traditional deep networks, either supervised or unsupervised, nowadays, DL owes its breakthrough to available large data storages and fast Graphics Processing Units (GPUs) with high computational power [8]. Open-source online libraries are available such as TensorFlow, Caffe, Theano, Torch, Deep Learning 4j (DL4j). Although none of these libraries are optimal, features, plus points, and drawbacks of each library, such as flexibility, speed, and integrability, should be considered to choose the most appropriate fitting library for the desired application [10, 11].

Common network architectures in DL include Multilayer Perceptron (MLP), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), autoencoders, and Generative Adversarial Networks (GANs). Although the true capacity of DL methods has not been revealed yet, each of them can address specific applications based on their architecture. For example, Deep Neural Networks (DNNs) have the potential to analyze multi-dimensional data and internal relationships among them, including but not limited to the prediction of protein structure and regulating gene expression. Moreover, RNNs can be used for sequential data sets where building blocks of data have a cyclic connection, such as Song Short-Term Memory Units (LSTMs), perceptrons, and Gated Recurrent Units (GRUs). Furthermore, CNNs are considered as the optimal fitting method for analyzing spatial data [10, 12].

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) are widely-used terms in modern data-driven research. Although these terms are closely related, it is critical to distinguish their differences for better use of methods belonging to each for a particular application. Hence, clarifying the differences between these terms worth mentioning. AI is the field of study to build intelligent systems, programs, and machines which can creatively solve problems. ML, a subset of AI, is the study of computer models and algorithms (e.g., neural networks), used by machines, to learn structures and patterns from observed data automatically, and then apply these learned patterns to make an inference on the rest of the unseen data. In classical ML, generally, three components are needed to learn patterns, namely datasets, features, and the algorithm. Therefore, selecting the appropriate features, depending on the application, plays a pivotal role in the success of the learning procedure, requiring domain expertise. DL, a subset of ML, is in fact a technique for realizing ML. For example, artificial neural networks (ANNs) are a type of DL algorithm that aims to imitate the way our brains make decisions. More specifically, an ANN is a web of layers, connections, and direction of data propagation to learn arbitrary functional mappings using data, resembling the functional structure of the human brain. ANNs can perform complex tasks such as decision making, cognition, patterns generating, and learning. In this regard, training aims to learn certain parameters of the ANN on a given learning task, which makes the feature selection process a part of the learning process. From this aspect, DL is a subfield of ML that enables an enhanced ability to find and amplify even the smallest patterns. DNNs were built by adding more hidden layers to ANNs, enabling the performance of more complex tasks by capturing nonlinear relationships. It is important to note that these models can be trained for both supervised and unsupervised learning tasks. Additionally, a combination of supervised and unsupervised ML methods for training DNNs is reported [9, 13-16].

2. Machine Learning Applications

In speech recognition, image processing, and complex control tasks, DL has proved to have superior performance compared to traditional ML methods and human perception [17, 18]. Furthermore, biomedicine and biomedical engineering, as data-rich disciplines, suffer from complex and often ill-understood data [19]. Contemporary microscopes, for instance, can provide up to 10^5 images per day,

which is challenging to be analyzed manually. ML methods can detect particular features, cluster, and classify images [20]. Analysis of cellular assays by ML methods has provided a unique opportunity for clinicians to detect, monitor, and remedy genetic perturbations [21, 22]. The performance of the ML methods is comparable to that of common image processing methods in the case of intricate multidimensional analysis such as distinguishing features.

2.1. Machine learning applications for assay quantification and classification

Inaccurate measurement can occur in the manual interpretation of colorimetric test results. The integration of ML with current methods not only can address the inaccuracy issue, but also can amplify the test speed. Colorimetric assays have been analyzed using ML. A machine learning-based mobile application was developed to quantify peroxide concentration using colorimetric test strips with almost 90% accuracy [23]. In another study, two different CNN models were used for assay type detection and colorimetric measurements: 8-layer deep AlexNet, consisting of series network, and 42-layer deep Directed Acyclic Graph (DAG) network of Inception v-3 [24]. Both methods had 100% accuracy in the training set for assay type detection with less than 1000 data samples needed for training of the algorithm, whereas the precision of colorimetric detection was low. Although the Inception method required virtually 20 times more time for training compared to the AlexNet, the former needed less memory and computational power. More time, larger training data set, and higher computational power were needed for more accurate results. Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), and ANN were used to achieve classification results for the detection of alcohol concentration in saliva [25]. LDA could aptly perform linear classifications when dataset classes are sufficiently separated from each other based on standard concentration values. SVM demands more training time because of the crucial optimization step. Notwithstanding the proposed optimizations, SVM's classification performance did not improve in comparison to LDA. Overall, ANN integrated with LAB color space presented the optimal performance. For spectral classification, Euclidean Distance (ED), Spectral Angle Mapper (SAM), SVM, Logistic Regression (Logi), and Multilayer Perceptron (MLP) models were employed for differentiating between seborrheic dermatitis and psoriasis on the scalp [26]. SVM yielded higher accuracy and sensitivity over other models. In another study, an ML method was used for single-molecule data analysis where CNN and SVM had 98.1% and 91.7% accuracy on test datasets, respectively [27]. A CNN-based model was developed with up to 19 layers to demonstrate the effect of depth on the algorithm's accuracy [28]. Augmenting a convolutional ConvNet depth enhances classification accuracy.

An ML method was implemented on a smartphone platform to automatically identify pH values [29]. Images of pH strips under different orientation and illumination conditions (Figure 3) and their colorimetric values were used as the training set for the Least Squares-Support Vector Machine (LS-SVM) classifier algorithms. LS-SVM had a 100% accuracy for all pH values. On the other hand, the SVM method yielded an inferior detection performance compared to LS-SVM, especially for 3, 6, 7, and 8 pH values.
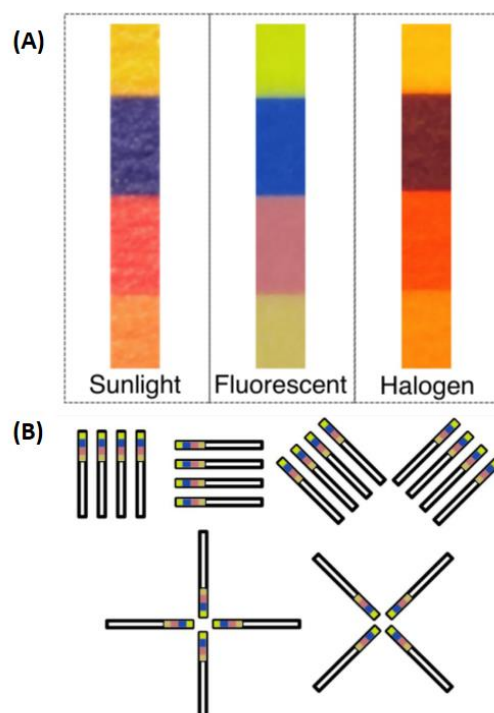
Figure 3. Colorimetric measurements of assays. (A) Influence of different illumination (sunlight - fluorescence – halogen light) and orientation on the obtained image of a strip with a pH of 12. (B) Different orientations proposed for pH strips to increase the variety of learning dataset for the algorithm [29]. Reproduced with permission from A. Y. Mutlu, V. Kılıç, G. K. Özdemir, A. Bayram, N. Horzum and M. E. Solmaz, Analyst **142** (13), 2434-2441 (2017). Copyright 2017 the Royal Society of Chemistry.

In another study, different classifiers were employed to analyze images to detect tuberculosis (TB) [30]. When 75% of the dataset was used for training and 25% for evaluating, the Bagged Tree classifier had the most optimal accuracy by 94.1% over other methods such as the Fine K Nearest Neighbor (KNN), with 70.6%, and the Cubic SVM, with 76.5%. However, when all the dataset was used as the training data, the optimal accuracy results can be changed as follows: the Bagged Trees 97.2%, the Fine KNN 94.4%, and the Cubic SVM 88.7%.

A mobile phone-based colorimetric test was developed by combining DL with a wax-printed paper-based multiplexed immunoassay to detect Lyme disease [31]. Training the algorithm with 48 positive and 52 negative samples, 90% sensitivity, 95% area under the curve, and 87% specificity was reported (Figure 4).

Figure 4. Sequential steps of the devised method to detect Lyme disease using a mobile phone to capture the image, and employing a DL approach to analyze the data to determine the test result [31]. Reproduced with permission from H.-A. Joung, Z. S. Ballard, J. Wu, D. K. Tseng, H. Teshome, L. Zhang, E. J. Horn, P.

M. Arnaboldi, R. J. Dattwyler, O. B. Garner, D. Di Carlo and A. Ozcan, ACS Nano 14 (1), 229-240 (2020).

2.2. *Microfluidic devices and machine learning*

Microfluidics can be used as a quick, low cost, and easy to use technique for the determination of biomarkers in clinical samples [32-42]. With the assistance of ML, these devices become more functional and accurate. ML-enabled microfluidic devices are broadly used in measuring fluidic properties in microfluidic devices [43], glucose assays [44], soft sensors [45], flow cytometry [46], and cytopathology [47-49]. NNs have been used to quantify physical properties in microfluidics. For example, the magnitude of the blend droplets was estimated [50], in which an ML algorithm was developed to predict emulsion stability in a microfluidic channel by learning the shape descriptor of the emulsion [51]. Briefly, microchannels were fabricated using Polydimethylsiloxane (PDMS) and stereolithography. By following the convolutional autoencoder architecture [52], an ML model was designed. This algorithm has a low-dimensional (8-dimensional) code to explain droplet patterns within a thick coating and anticipate if the particle becomes unsteady or breaks down from its exact shape.

A DNN was developed to calculate fluidic characteristics in a microfluidic setup learning from the surge of droplets in a microfluidic channel [43]. On-chip motion can be estimated using the Coriolis method [53], which detects the mechanical fluctuations based on thermal measurements and flow rate [54]. A thermometer and heater were used to evaluate the changes in liquid temperature. This approach required a visually translucent window and an outer camera as opposed to more complex methods described in other studies. In the microfluidic model, silicon oil and water-Isopropyl Alcohol Solutions (IPA) solutions were used as inputs (Figure 5). Images of the droplets produced at the crossing of the water-IPA solution were recorded with a broad-field magnifier. Images of droplets for an IPA at 5.5% water concentration for different flow rates are illustrated in figure 5c. A NN was equipped with 6000 droplet pictures (400 pictures for every flow measurement), with flow velocity distributing from (0.1 to 1.5) ml per hour with increments of 0.1 ml per hour, while the flow velocity of silicon oil was maintained at 2 ml per hour. 2900 new pictures were evaluated on the trained network (100 pictures for every flow rate distributing from 0.1 to 1.5 ml per hour with increments of 0.05 ml per hour). Figure 5d shows the results of DNN training for flow rate determination for all concentration values. The mean deviation of the anticipation was computed by taking the average value of the absolute relative variance between the anticipated value and the actual value. The average error for the trained model was 2.9% and 5.7% for new flow rates. As the flow regime transited based on the flow velocity, the DNN faced difficulty to anticipate new flow rates, with new flow regimes, which were not included in the training dataset.

Another DNN was equipped with 3600 pictures from each of the four-concentration distributing, from 4% to 7% with increments of 1%. The diagram in figure 5e contrasts the expected values with the ground facts. The average errors for the trained samples were 1.5% and 9.3% for new samples. Figure 5e shows that the DNN correctly predicted the concentrations of 4.5% and 6.5% but was unable to accurately predict the concentration of 5.5%. This inaccuracy suggested that the shape changes from 5.0% to 6.0% were non-monotonic. This non-monotonic manner was the output of a shift in the droplet flow shape between various conditions caused by a change in IPA concentration in the solution.
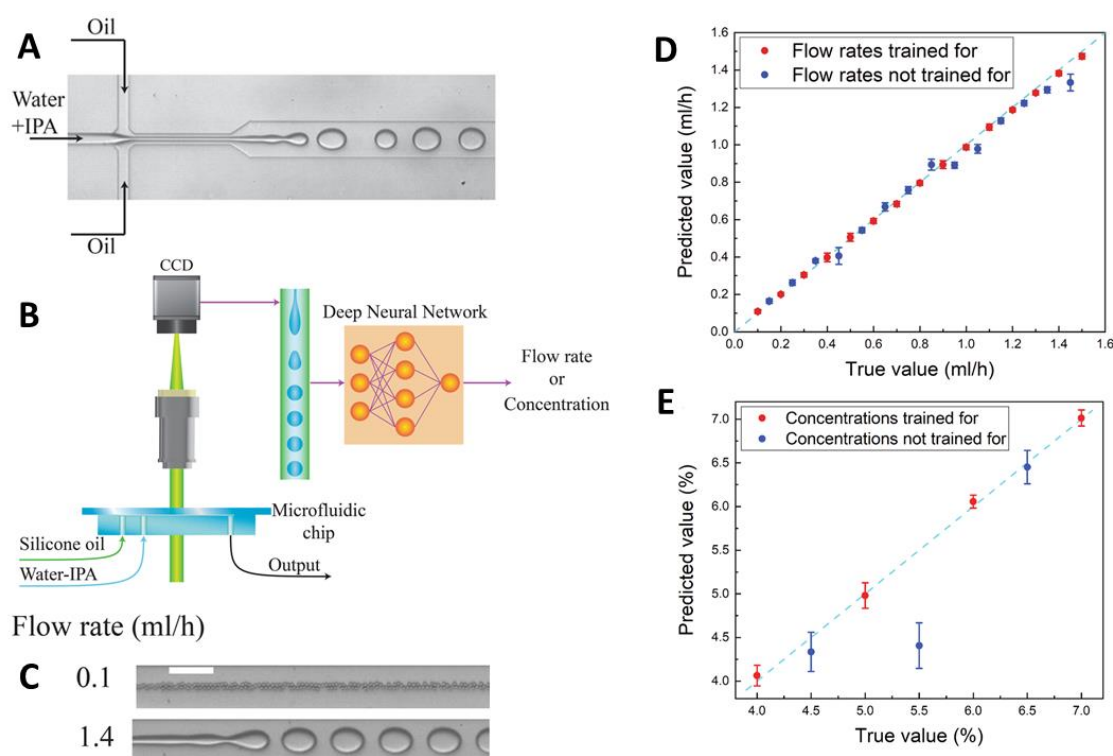
Figure 5. DNN applied to a microfluidic system. (A) Diagram of the microfluidic conduit. (B) The 2-stage structure of the creation of droplets in a microfluidic system carries data on flow characteristics. Algorithms were utilized to retrieve the characteristics while the actual method remains unchanged. (C) Photos of particle development at various flow speeds of the dissipated stage. (D) Flow velocity evaluation; estimated outcomes were compared to the actual value for 5.5% water-IPA concentration. Red dots indicate the flow velocity for which the system was trained, and the new flow velocities are shown in blue dots. (E) Comparison between anticipated concentration by DNN and actual data [43]. P. Hadikhani, N. Borhani, S. M. H. Hashemi and D. Psaltis, Sci. Rep. 9 (1), 1-7 (2019); licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.

Soft sensors can benefit from being integrated with ML methods. The soft sensor is a generic term for software, which processes multiple measurements simultaneously. Soft sensors are used to forecast response variables that are challenging to quantify. Soft sensors have been formulated from carbon particles[55, 56], silver nanowires [57, 58], ambient-temperature molten metals [59, 60], and ion liquids [61, 62]. The two common disadvantages of soft sensors correlated to conventional sensors are the hysteresis and deviation in response, mostly shown by microfluidic soft sensors [60, 63]. Another drawback is the placement of signal wires when there is limited space compared to the number of sensors. A hierarchical recurrent sensing network model, a type of RNNs, was developed to solve the above issues in soft sensors. Two different pressure sensors (a) a straight channel with three distinct cross-section areas in three different parts, (b) a similar-sized channel with distinct curved shapes (square, triangular, and circular) were manufactured with one microchannel poured with liquid material (eutectic gallium-indium or EGaIn) [45]. 2070 training data and 375 test datasets were prepared for each sensor by varying the pressure and pressing speed in different locations. The test accuracy of these devices was measured in different places for different values of pressure. The overall Normalized Root Means Squared Error (NRMSE) was 6.64%, and localization accuracy was 81.87% for straight channel. The NRMSE of similar-sized channel model was 5.81%, and the accuracy of localization was 85.42% overall of the test cases. It was reported that the implementation of RNN decreased the number of needed signal wires in a soft sensor array and simplified the calibration process.

Another study investigated the potential of hierarchical feature extraction methods for the automatic design of a sequence of pillars to obtain user-defined fluid deformation in microfluidic

devices. The accuracy and needed time were compared for the deep convolutional neural network (CNN), pre-trained deep neural network (DNN), and genetic algorithm (GA). The proposed DNN was comprised of 1000 hidden units in 5 hidden layers, while the CNN consisted of 2 convolutional layers, 2 pooling layers, and one fully connected layer with 500 hidden units. 150,000 samples were used to train the NNs, and 20,000 additional samples were used for validation. Comparing the run time and pixel match rate (PMR) of the GA and deep learning methods, although the PMR of GA had a higher performance than DL, the runtime of GA can be 600 times more than the needed time for DL [64].

Regardless of the selected manufacturing method, a prevalent problem of microfluidic devices, in the production stage, is the inability of the fabrication approach to reproduce exactly the same dimensions, resulting in the inconsistency of the flow rate and the obtained results. Deep Q-network (DQN) and model-free episodic controllers (MFEC), as two reinforcement learning methods, are used to maintain stable flow conditions over an extended time period, with minimum need for manual intervention, to ensure consistent result acquisition. This was obtained by observing the microfluidic chip by a camera and analyzing the data by image processing. The performance of the DQN algorithm was comparable to that of human testers after 37h of training of the DQN with more than 200,000 image frames. More training increased the performance of DQN over human tester. On the other hand, MFEC reached its peak performance in 2 h of training with 11,000 image frames, which was considerably fast compared to DQN that needed 24 hours (130,000 image frames) to yield the analogous accuracy level. However, the maximum performance of MFC could not surpass the performance of the human tester [65].

2.3. Biomedical applications of machine learning

Biology has great potential to be integrated with ML approaches to address unmet needs. Sorting sperms by ML, for instance, is one of these applications [66]. Assessing the morphology, motility, and concentration of sperms in semen can provide important information for fertility clinicians. Although the point of care approaches for sperm motility and concentration measurements are available, evaluation of sperm morphology in point of need has not been developed to its potential, due to the time-consuming process [67]. In vitro fertilization (IVF) is a commonly used method which relies on the selection of sperms with suitable morphology. Since sperms start to die after being out of the body for a while, the selection process should be done as fast as possible. Utilizing differential interference contrast microscopy by clinicians, as one of the promising selection methods, not only is a time-consuming process, but it also is prone to be affected by clinicians' bias and inexperience. However, despite being laborious and bias-prone, manual sperm morphology assessment by human clinicians is still the most prevalent in-use methodology since other proposed alternative approaches are expensive as well as inaccurate. ML approaches can significantly amplify both accuracy and rapidness. A recent study reported 88.67% area under the accuracy recall curve and 90% accuracy using the SVM classifier to select desired sperms automatically [68]. Since enough number of all existing types of sperms may be unavailable, sperm head images were categorized by using transfer learning on a Visual Geometry Group (VGG16) CNN [69]. Firstly, the model was trained by the ImageNet database, a user-annotated database comprised of images of typical objects and animals. Subsequently, sperm head images were used to train the classifier. Overall, the accuracy of 94.1% and 62% were reported for the Human Sperm Head Morphology dataset (HuSHeM) and partial-agreement laboratory for Scientific Image Analysis Gold-standard for Morphological Sperm Analysis (SCIAN) datasets, respectively, which outperformed conventional ML methods such as Centered Hyperellipsoidal Support Vector Machine (CE-SVM) [67]. A transfer learning method was applied on a deep CNN in which 80% of 3820 sperm images were used to train the algorithm, while the developed algorithm was validated by the remaining portion of the data set. The trained algorithm was able to correctly identify 371 sperm images out of 415 images, yielding an accuracy of 89% based on annotations obtained from clinicians [70].

Another application of ML methods in the realm of biology is in cell detection. Four major types of white blood cells were detected using the Residual Network (ResNet) V1 50 DL algorithm with

100% accuracy [71]. In another study, a weakly supervised DL architecture outperformed VGG and Microtubule Networks (MT) for detecting and counting dead cells in microscopy images [72]. ML was employed for single-molecule data analysis, where CNN and SVM had 98.1% and 91.7% accuracy on test datasets, respectively [27]. Detecting diseases and disorders is another application of ML in biology. Random Forecast (RF) ML has been utilized for the classification of the most common neurodevelopmental disorder Attention-deficit and Hyperactivity Disorder (ADHD) with 82% accuracy, 75% sensitivity, and 86% specificity [73]. 91.6% accuracy was achieved in analyzing electronic medical data for detecting children of severe hand, foot, and mouth disease (HFMD) using ML [74]. Furthermore, a method was developed for cytopathological photo review, employing ML in microfluidic devices [47]. This study explored the validity of using DL algorithms for cytopathological research by classifying three major unlabeled, unstained cell lines of leukemia (MOLT, HL60, and K562). By using restricted Boltzmann machines, a deep belief network [75] was developed, in which the scales were adjusted to discover a common abstract depiction of the data framework without considering the names. Moreover, a microfluidic cytometer was created based on contact-imaging with ML for high-performance development by a single frame [46]. In this work, a high-performance single-frame growing with in-line ML was developed for cell interact pictures. A similar model of microfluidic cytometer-based touch imagery was demonstrated for cell recognition and calculation. A serious challenge of diagnosing AIDS-related cancers (e.g., diffuse large B cell lymphoma (DLBCL)) is the lack of comprehensive tests and classification, particularly in deprived regions. An automated, portable, robust, and cost-effective digital cellular analysis test was developed, in which a DNN processed the data to provide quantitative result readouts, including cell size, malignant cell number, and differentiation between high/low-grade subtypes. The device could be used while connected to the Internet or based on an installed Raspberry Pi processor in remote areas. The proposed DNN was trained by 3447 training data and validated by 1732 samples. Using the proposed DL technique, the needed time for computation reduced 5 times compared to image reconstruction of the whole field of view. The proposed device was reported to have 91% sensitivity, 100% specificity, and 95% accuracy for diagnosing lymphoma[76].

Employing ML approaches in paper-based devices has become more ubiquitous recently. Paper is a useful medium for microfluidic assays as it is lightweight, low cost, compatible with biological objects, easy to transport and store. Other practical materials such as yarn and fabric have been also employed in creating microfluidic devices [77-83]. Yarn and thread are promising resources for microfluidic devices due to their biochemical properties. These devices include a microfluidic paper/yarn-based analytical device (µTPAD) and 3D Microfluidic Paper-based Analytical Devices (µPAD) [44]. These devices exhibited the viability of using fitting and classification algorithms for an ANN to derive glucose concentration based on dye data from four channels of Cyan Magenta Yellow Black (CMYK). To train and test the ANN, mean 16-bit color values were obtained in a device from all the four-color paths in the CMYK chart. The data utilized to prepare the ANN was comprised of 160 data points for µPAD, and 54 data points for µTPAD, where two different methods (fitting and classification) were performed. Figure 6a reflects the efficiency of ANNs applied to the fitting problem. The classification accuracy of 91.2% and 94.4% were reported for µPAD analysis sites and the µTPAD, respectively.
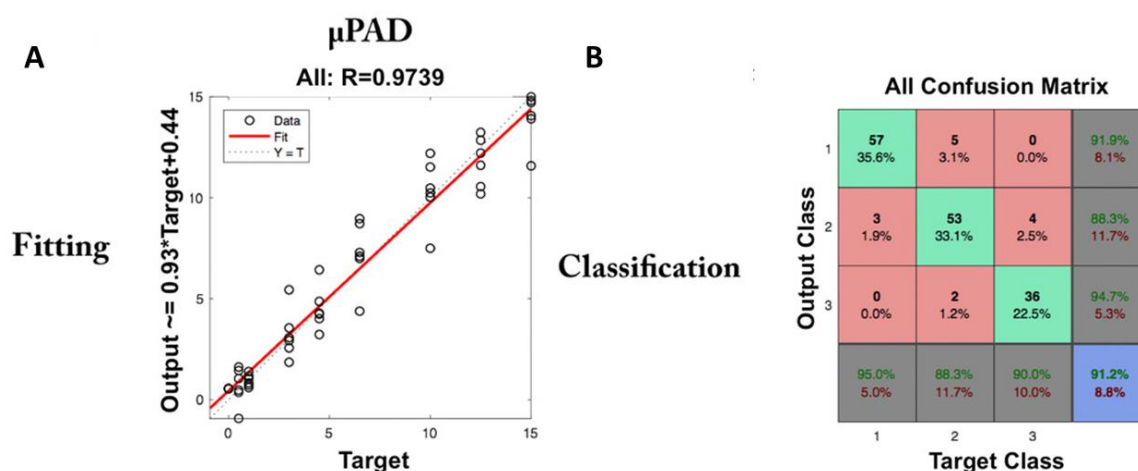
Figure 6. ML in paper-based microfluidics. (A) Generated standardization lines from regression data utilizing the neural network fitting model. Every dot on the graph shows a forecast anticipated by the model. The x-axis marked as "Target" signifies the glucose concentration at the experiment site, while the y-line classified as 'Output' implies the glucose concentration prediction made by the model. (B) An error matrix classification model. This figure displays a 3×3 error matrix, where the rows indicate the three "output" groups where the assessment site may be held by the ANN, and the columns signify the respective true or "Target" group of the assessment site. For instance, the "53" in the second row and second column matrix cell shows that 53 assessment sites affiliated to group 2, and they were properly classified as group 2 sites by the ANN classifier. Also, the "5" in the first row and second column cell in the matrix shows that three assessment sites affiliated to group 2 were misplaced as group 1 sites by the ANN classifier [44]. Reproduced with permission from W. Lee, A. Gonzalez, P. Arguelles, R. Guevara, M. J. Gonzalez-Guerrero and F. A. Gomez, Electrophoresis **39** (12), 1443-1451 (2018). Copyright 2018 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.

Computation holds great potential in diagnostics, where the computational sensing methods will advance point-of-care (POC) analysis. To evaluate the signs generated on paper-based substrates, ML algorithms can be used in POC sensors. A Paper-based Vertical Flow test (VFA) was created using ML for cheap and rapid high-sensitivity C-reactive protein (hsCRP) testing [84]. First, a multiplexed VFA platform was developed using paper sheets piled inside a 3D-printed frame, designed to sustain a uniform vertical serum flow through a sensing membrane of 2D nitrocellulose (NC) (Figure 7a). The ultimate CRP quantification model was developed using 209 training set and the best spot framework. Next, blind evaluation utilizing the configured hsCRP VFA framework and qualified model was performed with 57 test samples. The samples were analyzed using the pixel information that had 28 spots and 5 conditions inside the computationally defined subset. The model accomplished 100% accuracy for classification by adequately classifyng 6 specimens as acute, and the remaining 51 specimens as in the hsCRP range. A comparison of the VFA quantification accuracy with the gold standard values is shown in figure 7c, d, demonstrating acceptable agreement in the term of quantification precision. The R^2 value of the system was 0.95, with a linear best-fit line slope at 0.98 and intercepted at 0.074.
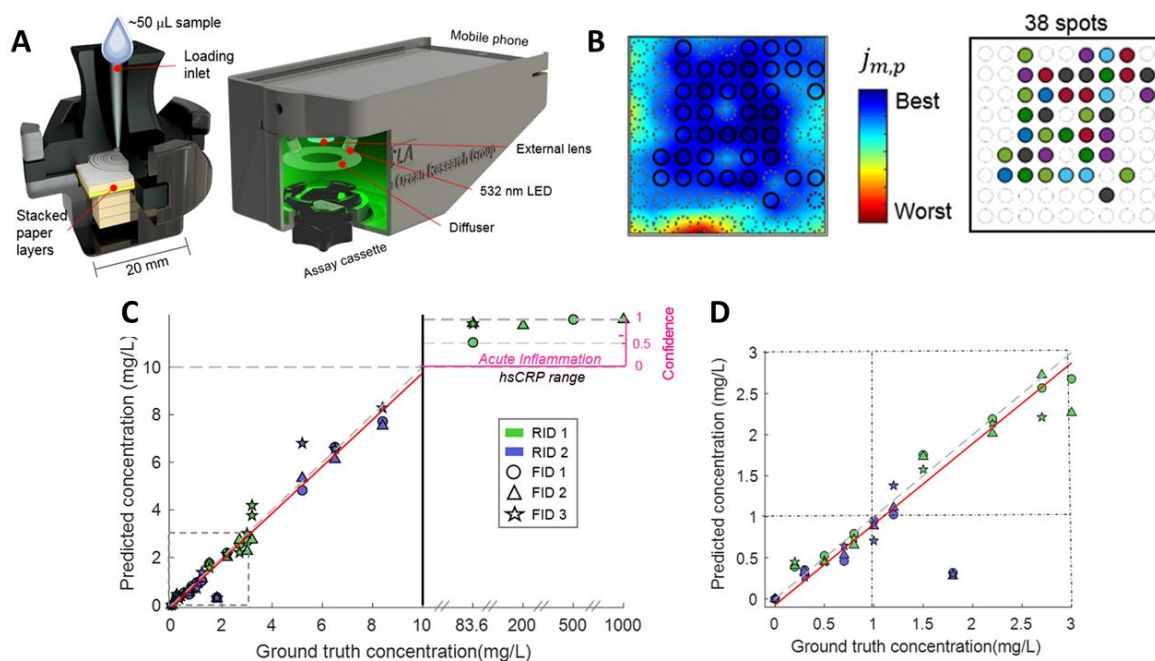
Figure 7. Application of ML methods in vertical flow assays. (A) The cross-sectional view of the VFA cassette. (B) The screening process for the point. Plotting the cost function throughout the sensing element generates a heat-map (left), and the ideal subgroup of points (right) is then chosen for the most appropriate quantification effectiveness of the deep red indicator. (C) Blindly tested clinical specimens, where the left y-axis indicates the VFA predicted the concentration of CRP and the right y-axis indicates the confidence score for the specimens categorized as acute, and the x-axis presents the ground truth concentration. The actual match (y=x) is represented by a dotted line, and the linear fit is indicated by the red line. Reagent batch ID and fabrication batch ID are indicated by the marker color and patterns. (D) Blind testing results for moderate and low CVD risk areas, where the dotted line indicates the clinical cutoffs [84]. Z. S. Ballard, H.-A. Joung, A. Goncharov, J. Liang, K. Nugroho, D. Di Carlo, O. B. Garner and A. Ozcan, NPJ digital medicine 3 (1), 1-8 (2020), licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.

3. Implementation challenge

3.1. I. Required number of samples for the desired accuracy

After training an algorithm, if the trained model could accurately perform prediction on the training dataset as well as an independent dataset, the training can be deemed complete [7]. Measuring the accuracy of a learner algorithm plays a pivotal role in designating a robust method. One of the common means to attain this goal is to measure the ratio of erroneously classified samples to the total number of samples [5]. Other ratios such as "standard false positive" and "false negative" rates can be applied to define the meticulousness of an algorithm [85]. However, in some cases, a common problem with training and testing of the algorithms happens when the training error is subtle, but conversely, the test error is considerable [86]. This indicates that the algorithm fails to generalize the structure in independent data properly (overfitting). On the other end of the continuum, when an algorithm fails to predict training data after learning, under-fitting occurs [7, 87]. These two problems are the major issues with the unsatisfactory performance of algorithms. The underlying cause of overfitting is the complexity of the model. This phenomenon occurs when the number of adjustable parameters is proportionally more than the number of training samples. In contrast, under-fitting stems from the simplicity of the algorithm. In this regard, the former can be addressed by augmenting the number of training samples or moderating parameters, where the latter can be solved by increasing the complexity of the model [88].

Taking these problems into account, finding the minimum number of required data to train a precise algorithm is of great importance [89]. However, there is no rule of thumb for determining this number since it is highly dependent on the method, number of classes, and quality of data,. [90]. In general, the more the number of features in data is, the more training data is needed [91]. Besides, generative methods require fewer training samples in comparison to discriminative models [92]. To elucidate this approach, an experiment was carried out on tumor cells [48]. Figure 8 shows that increasing the number of training samples improves the model performance by decreasing the test error, and amplifies the training error, because the algorithm fails to fit several data points. Hence, the sample amount is a trade-off between test error and training error. For instance, in this specific test, 850 training samples could be enough since more samples cannot converge two curves any further.
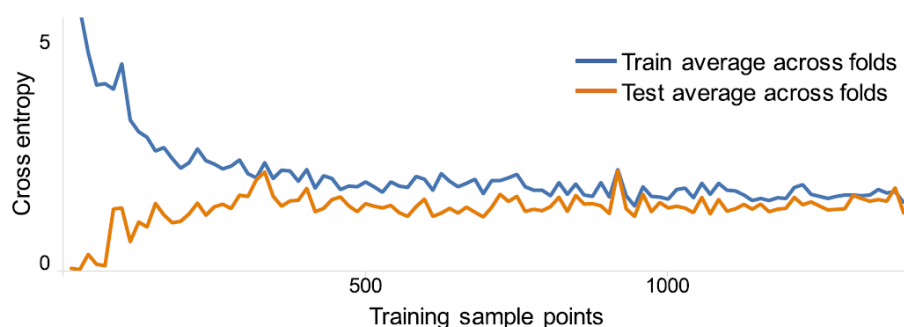

Figure 8. The deviation between test and train error diminishes up to N≈850. Thereafter, augmenting training samples would not decrease this discrepancy considerably. Hence, the minimum required data for this experiment is N≈850 [48]. C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi and B. Jalali, Sci. Rep. 6, 21471 (2016), licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.

3.2. II. Dealing with data-limited cases

DL demands a substantial amount of data for superior accuracy. Clinical medicine, however, is resource-restricted since a limited number of patients and clinical records are present in the time and place of training [19]. To address this, Electronic Health Record (EHR) data can be a promising solution [93]. EHR is a digital record of patients' health information, including personal statistics, laboratory test results, medical history, Magnetic Resonance Imaging (MRI), and Computerized Tomography (CT) scan images, available for authorized users worldwide [94, 95]. Another effective method of dealing with data-limited experiments is augmenting available data. Rich data such as 3D images can be divided into lower dimension images to train algorithms. For instance, training a CNN by 2.5D data for CT scan image detection yields almost analogous precision performance with 3D trained CNN [96]. Furthermore, taking images of available data under different illumination conditions with various orientations can increase the training data set (Figure 3b) [29]. However, in some cases neither dividing into lower dimensions nor different illumination and orientation is feasible. In such cases, transfer learning can assist experts to overcome this problem. Transfer learning means training an algorithm on a set of data and taking the advantages of the algorithm on entirely different data [24]. Transfer learning was applied in human sperm classification with an accuracy of 94.1% for the HuSHeM dataset [69]. However, the accuracy of a CNN trained on a limited number of labeled MRI can outperform a classifier trained on a large dataset from a dissimilar domain [97].

3.3. III. Image acquisition

The detection of color could be difficult since a diversity of factors could affect the interpretation of data, including the illumination intensity and its direction, ambient lighting conditions, as well as the

employed camera's features. Mercury and xenon lights, which are widely in use, provide a varying lighting intensity based on their lifetime and heating-up period. Nevertheless, contemporary Light-emitting Diodes (LEDs) and Surface Mount Device (SMDs) provide stable light intensity, applicable to ML approaches [5]. For early experiments conducted in colorimetric tests, mostly, flatbed scanners were used to capture images from samples [98]. Although this approach can eradicate the problem with varying illumination conditions and the constant distance between the sample and camera [99], the major downsides with these scanners not only were their inaccessibility, but they also might not be utilized in experiments with liquid or wet samples [100]. In contrast, smartphone-based experiments have surmounted this problem by providing portable cameras without the need for the direct contact of the camera and the sample during the image taking process. Another issue with wet or liquid samples is the reflection of light from their surface, which can be misleading data for an algorithm trained on dry samples for the same experiment, namely pH detection [101]. Furthermore, the shape and properties of the sample container should not be neglected in experiments with liquid samples since the shadow of edges and transparency of the container can affect the color of the sample taken with smartphones [24].

RAW images, by which specialists chiefly mean "unprocessed" images, comprise of original information outcomes from a camera's lens, in 10-14 bits of color information, before in-camera processes [102]. Whereas, Joint Photographic Experts Group (JPEG) images are compressed, small size files with only 8 bits of color depth [103]. This compression triggers concern in experts regarding the suitability of JPEG images in image processing [104]. In spite of this concern, using LS-SVM as an ML classifier, JPEG format had an analogous performance with the RAW format for peroxide content quantification [29].

4. Future Prospects

Further applications of ML are conceivable in a large scope from Lab on a chip (LoC) to Structural Health Monitoring (SHM) [105], i.e. airplane, bridges, skyscrapers health monitoring by deciphering data from several sensors on them [106]. Most of the experiments employ ML and DL for post-experiment data analysis. ML may not play a decisive role in the design and control of the experiment. However, learning from previous experiments, AI can determine the optimized proportion of reagents and samples in microfluidic tests, as well as the best time for injecting them [8]. Furthermore, DL algorithms can assess the design aspect of novel proposed devices to determine the most efficient design by taking into account experimental material, the required time for the fabrication of each design, price, and efficacy of reactions.

Another field which can benefit from ML and DL is organ-on-a-chip (OOC) systems. Artificial tissues could to be mimicked in laboratories for a diversity of applications such as the replacement of organs in the body, regulatory drug testing, and experimental disease monitoring [107, 108]. Culturing, maintaining, and monitoring on-chip tissues will generate a substantial number of images and videos of living cells, tissues, organs in the in-vitro environment, and the effect of drugs on them [109]. Such big data needs to be analyzed from spatial and histological, aspects to evaluate expected features of OOC systems. Hence, the design, control, and self-regulation of OOC systems is a feasible future prospect for DL.

Paper-based microfluidic tests, as a low cost and accessible method, will be in use more ubiquitously in the future [110]. By distributing these devices globally, collecting data, and analyzing them using DL will bring an unprecedented opportunity to detect symptoms of illnesses [111], malnourishment in certain societies or regions, and predicting outbreaks of diseases. For instance, Zika Virus could be detected by a paper-based sensor [112]. Overall, monitoring the health status of a large portion of the world population by POC devices and DL can facilitate control and overcome pandemics.

Microorganisms affect planet earth and humanity with an unneglectable role in climate change, oxygen supply, and carbon cycles [113]. Thus, monitoring their trends and the influence of global warming on them require advanced environmental microfluidic monitoring technologies for testing their concentration in soil and oceans[114]. The copious amount of information gathered by LoC devices

may require DL to collect, classify, quantify, and analyze the data for later uses in making a decision regarding how to control climate change.

5. Discussions and Conclusions

To choose the best method, the available data should be considered. If enough number of labeled input and output data is available, then the supervised method can be the best choice. Otherwise, if the number of annotated input data is more than labeled output data, then semi-supervised methods can be chosen. Finally, if output data is not labeled at all, the unsupervised method should be considered. Since ML uses NNs that resemble the human neural system, algorithms may learn more efficaciously from a certain format of representing data compared to other formats. Featurization, which is the process of converting raw data into an appropriate input format, has recently attracted attention [87]. According to the algorithms' ability to handle different types of data, each of them has its own applications. For instance, for cell biology applications, SVM can be intriguing [115, 116]. Moreover, linear discriminant analysis, generative approaches in general, attracted more attention for classifying the phenotypes of the actin cytoskeleton in Drosophila melanogaster cells [117]. The integration of existing DL algorithms allows the production of more capable architectures. Combining CNNs and RNNs, for example, has resulted in an algorithm that can be used for captioning images, summarizing videos, and image-question answering [10]. Therefore, combinations that are capable of executing more complicated tasks should be developed.

Regarding biomedical applications, for instance, disease evolution and symptoms of the known diseases can vary from person to person. Thus, even if an algorithm, trained by data from a restricted database, has an acceptable performance currently, there is no guarantee that it can adequately and reliably perform its task under new circumstances. Moreover, the issue of limited available samples can be solved in some colorimetric applications. Nevertheless, some biomedical fields have a limited number of ill people that are willing to participate in clinical research [90]. Hence, a global EHR platform can be created for gathering all available samples globally.

The main challenge in ML is the "black-box" issue [90]. ML algorithms are comprised of numerous hidden layers. Although these algorithms are developed by humans, the exact procedure of analyzing input data and the underlying logical reason behind the decision of ML inside these hidden layers are not fully understood. In some applications, such as annotating images and voice recognition, the user can instantly verify the outcome of the ML algorithm to ensure the quality as well as the accuracy of the result. However, the black-box issue brought about some predicaments in multi-dimensional applications. These applications inextricably associate patients' health in which the ML method is supposed to determine the dosage of each constituting component of the drug based on the symptoms of the patient as the input data. Since it is not transparent how the ML algorithm reaches the final arrangement of drug elements, it causes a dilemma for both experts and patients: whether an expert should trust the suggested drug as the end product; and the patient would be willing to use prescriptions of ML architectures [4]. Different ML methods may yield different results for the same input data, augmenting this uncertainty [86, 118].

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

1. A. K. Waljee and P. D. R. Higgins, Am. J. Gastroenterol. **105** (6), 1224-1226 (2010).

2. D. J. Williamson, G. L. Burn, S. Simoncelli, J. Griffié, R. Peters, D. M. Davis and D. M. Owen, Nat. Commun. **11** (1), 1-10 (2020).

3. A. L. Samuel, IBM J. Res. Dev. **3** (3), 210-229 (1959).

4. J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah and M. Spitzer, Nat. Rev. Drug Discovery **18** (6), 463-477 (2019).

5. C. Sommer and D. W. Gerlich, J. Cell Sci. **126** (24), 5529-5539 (2013).

6. C. Sommer, R. Hoefler, M. Samwer and D. W. Gerlich, Mol. Biol. Cell, 134627 (2017).

7. D. M. Camacho, K. M. Collins, R. K. Powers, J. C. Costello and J. J. Collins, Cell **173** (7), 1581-1592 (2018).

8. J. Riordon, D. Sovilj, S. Sanner, D. Sinton and E. W. Young, Trends Biotechnol. **37** (3), 310-324 (2019).

9. J. T. Senders, P. C. Staples, A. V. Karhade, M. M. Zaki, W. B. Gormley, M. L. D. Broekman, T. R. Smith and O. Arnaout, World Neurosurg. **109**, 476-486.e471 (2018).

10. S. Min, B. Lee and S. Yoon, Brief. Bioinform. **18** (5), 851-869 (2016).

11. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean and M. Devin, arXiv preprint arXiv:.04467 (2016).

12. I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*. (MIT press, 2016).

13. K. G. Liakos, P. Busato, D. Moshou, S. Pearson and D. Bochtis, Sensors **18** (8), 2674 (2018).

14. T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis and K. C. Chatzisavvas, Simulation Modelling Practice and Theory **55**, 1-9 (2015).

15. D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo and G.-Z. Yang, IEEE journal of biomedical and health informatics **21** (1), 4-21 (2016).

16. J. Schmidhuber, Neural Networks **61**, 85-117 (2015).

17. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam and M. Lanctot, Nature **529** (7587), 484 (2016).

18. G. Saon, H.-K. J. Kuo, S. Rennie and M. Picheny, arXiv preprint arXiv (2015).

19. T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, W. Xie, G. L. Rosen, B. J. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. E. Carpenter, A. Shrikumar, J. Xu, E. M. Cofer, C. A. Lavender, S. C. Turaga, A. M. Alexandari, Z. Lu, D. J. Harris, D. DeCaprio, Y. Qi, A. Kundaje, Y. Peng, L. K. Wiley, M. H. S. Segler, S. M. Boca, S. J. Swamidass, A. Huang, A. Gitter and C. S. Greene, J. R. Soc. Interface **15** (141), 20170387 (2018).

20. C. Conrad and D. W. Gerlich, J. Cell Biol. **188** (4), 453-461 (2010).

21. W. J. Godinez, I. Hossain and X. Zhang, BioRxiv, 361410 (2018).

22. S. Finkbeiner, M. Frumkin and P. D. Kassner, Neuron **86** (1), 160-174 (2015).

23. M. E. Solmaz, A. Y. Mutlu, G. Alankus, V. Kılıç, A. Bayram and N. Horzum, Sensors Actuators B: Chem. **255**, 1967-1973 (2018).

24. M. H. Tania, K. T. Lwin, A. M. Shabut, K. J. Abu-Hassan, M. S. Kaiser and M. A. Hossain, presented at the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2019 (unpublished).

25. H. Kim, O. Awofeso, S. Choi, Y. Jung and E. Bae, Appl. Opt. **56** (1), 84-92 (2017).

26. S. Kim, J. Kim, M. Hwang, M. Kim, S. J. Jo, M. Je, J. E. Jang, D. H. Lee and J. Y. Hwang, Biomedical optics express **10** (2), 879-891 (2019).

27. T. Albrecht, G. Slabaugh, E. Alonso and S. M. R. Al-Arif, Nanotechnology **28** (42), 423001 (2017).

28. K. Simonyan and A. Zisserman, arXiv preprint arXiv (2014).

29. A. Y. Mutlu, V. Kılıç, G. K. Özdemir, A. Bayram, N. Horzum and M. E. Solmaz, Analyst **142** (13), 2434-2441 (2017).

30. K. J. AbuHassan, N. M. Bakhori, N. Kusnin, U. Z. Azmi, M. H. Tania, B. A. Evans, N. A. Yusof and M. Hossain, presented at the 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC), 2017 (unpublished).

31. H.-A. Joung, Z. S. Ballard, J. Wu, D. K. Tseng, H. Teshome, L. Zhang, E. J. Horn, P. M. Arnaboldi, R. J. Dattwyler, O. B. Garner, D. Di Carlo and A. Ozcan, ACS Nano **14** (1), 229-240 (2020).

32. B. Yenilmez, S. Knowlton and S. Tasoglu, Adv. Mater. Technol. **1** (9), 1600144 (2016).

33. S. Knowlton, C. H. Yu, N. Jain, I. C. Ghiran and S. Tasoglu, PLoS One **10** (8), e0134400 (2015).

34. B. Yenilmez, S. Knowlton, C. H. Yu, M. M. Heeney and S. Tasoglu, Adv. Mater. Technol. **1** (5), 1600100 (2016).

35. R. Amin, S. Knowlton, B. Yenilmez, A. Hart, A. Joshi and S. Tasoglu, RSC advances **6** (96), 93922-93931 (2016).

36. R. Amin, S. Knowlton, J. Dupont, J. S. Bergholz, A. Joshi, A. Hart, B. Yenilmez, C. H. Yu, A. Wentworth and J. J. Zhao, J. 3D Print. Med. **1** (3), 155-164 (2017).

37. S. Knowlton, A. Joshi, P. Syrrist, A. F. Coskun and S. Tasoglu, Lab Chip **17** (16), 2839-2851 (2017).

38. S. Knowlton, C. H. Yu, F. Ersoy, S. Emadi, A. Khademhosseini and S. Tasoglu, Biofabrication **8** (2), 025019 (2016).

39. S. Knowlton, I. Sencan, Y. Aytar, J. Khoory, M. Heeney, I. Ghiran and S. Tasoglu, Sci. Rep. **5**, 15022 (2015).

40. E. Lepowsky, F. Ghaderinezhad, S. Knowlton and S. Tasoglu, Biomicrofluidics **11** (5), 051501 (2017).

41. S. Knowlton, A. Joshi, B. Yenilmez, I. T. Ozbolat, C. K. Chua, A. Khademhosseini and S. Tasoglu, International Journal of Bioprinting **2** (2) (2016).

42. S. M. Knowlton, M. Sadasivam and S. Tasoglu, Trends Biotechnol. **33** (4), 221-229 (2015).

43. P. Hadikhani, N. Borhani, S. M. H. Hashemi and D. Psaltis, Sci. Rep. **9** (1), 1-7 (2019).

44. W. Lee, A. Gonzalez, P. Arguelles, R. Guevara, M. J. Gonzalez-Guerrero and F. A. Gomez, Electrophoresis **39** (12), 1443-1451 (2018).

45. S. Han, T. Kim, D. Kim, Y.-L. Park and S. Jo, IEEE Robotics and Automation Letters **3** (2), 873-880 (2018).

46. X. Huang, J. Guo, X. Wang, M. Yan, Y. Kang and H. Yu, PLoS One **9** (8) (2014).

47. G. Gopakumar, K. H. Babu, D. Mishra, S. S. Gorthi and G. R. S. Subrahmanyam, JOSA A **34** (1), 111-121 (2017).

48. C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi and B. Jalali, Sci. Rep. **6**, 21471 (2016).

49. D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley and M. W. Covert, PLoS Comp. Biol. **12** (11) (2016).

50. Y. Mahdi and K. Daoud, J. Dispersion Sci. Technol. **38** (10), 1501-1508 (2017).

51. J. W. Khor, N. Jean, E. S. Luxenberg, S. Ermon and S. K. Tang, Soft Matter **15** (6), 1361-1372 (2019).

52. V. Turchenko, E. Chalmers and A. Luczak, arXiv preprint arXiv:1701.04949 (2017).

53. T. Wang and R. Baker, Flow Measurement and Instrumentation **40**, 99-123 (2014).

54. C.-Y. Lee, C.-H. Lin and Y.-M. Lo, Sensors **11** (4), 3706-3716 (2011).

55. J. Muth, D. Vogt and R. Truby, Embedded 3D printing of strain sensors within highly stretchable elastomers," Adv. Mater **26**, 6307-6312 (2014).

56. M. Amjadi, Y. J. Yoon and I. Park, Nanotechnology **26** (37), 375501 (2015).

57. M. Amjadi, A. Pichitpajongkit, S. Lee, S. Ryu and I. Park, ACS Nano **8** (5), 5154-5163 (2014).

58. M. D. Ho, Y. Ling, L. W. Yap, Y. Wang, D. Dong, Y. Zhao and W. Cheng, Adv. Funct. Mater. **27** (25), 1700845 (2017).

59. Y.-L. Park, B.-R. Chen and R. J. Wood, IEEE Sens. J. **12** (8), 2711-2718 (2012).

60. Y.-L. Park, D. Tepayotl-Ramirez, R. J. Wood and C. Majidi, Appl. Phys. Lett. **101** (19), 191904 (2012).

61. J.-B. Chossat, Y.-L. Park, R. J. Wood and V. Duchaine, IEEE Sens. J. **13** (9), 3405-3414 (2013).

62. J.-B. Chossat, H.-S. Shin, Y.-L. Park and V. Duchaine, Journal of Mechanisms and Robotics **7** (2) (2015).

63. Y.-L. Park, C. Majidi, R. Kramer, P. Bérard and R. J. Wood, Journal of Micromechanics and Microengineering **20** (12), 125029 (2010).

64. K. G. Lore, D. Stoecklein, M. Davies, B. Ganapathysubramanian and S. Sarkar, presented at the Feature Extraction: Modern Questions and Challenges, 2015 (unpublished).

65. O. J. Dressler, P. D. Howes, J. Choo and A. J. deMello, ACS omega **3** (8), 10084-10091 (2018).

66. C. McCallum, J. Riordon, Y. Wang, T. Kong, J. B. You, S. Sanner, A. Lagunov, T. G. Hannam, K. Jarvi and D. Sinton, Communications Biology **2** (1), 250 (2019).

67. P. Thirumalaraju, M. K. Kanakasabapathy, C. L. Bormann, H. Kandula, S. K. S. Pavan, D. Yarravarapu and H. Shafiee, Fertil. Steril. **112** (3), e41 (2019).

68. S. K. Mirsky, I. Barnea, M. Levi, H. Greenspan and N. T. Shaked, Cytometry A **91** (9), 893-900 (2017).

69. J. Riordon, C. McCallum and D. Sinton, Computers in biology medicine **111**, 103342 (2019).

70. P. Thirumalaraju, C. Bormann, M. Kanakasabapathy, F. Doshi, I. Souter, I. Dimitriadis and H. Shafiee, Fertil. Steril. **110** (4), e432 (2018).

71. M. Habibzadeh, M. Jannesari, Z. Rezaei, H. Baharvand and M. Totonchi, presented at the Tenth International Conference on Machine Vision (ICMV 2017), 2018 (unpublished).

72. S. Chen, A. Li, K. Lasick, J. Huynh, L. Powers, J. Roveda and A. Paek, presented at the 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019 (unpublished).

73. A. Kautzky, T. Vanicek, C. Philippe, G. Kranz, W. Wadsak, M. Mitterhauser, A. Hartmann, A. Hahn, M. Hacker and D. Rujescu, Translational psychiatry **10** (1), 1-9 (2020).

74. G. Liu, Y. Xu, X. Wang, X. Zhuang, H. Liang, Y. Xi, F. Lin, L. Pan, T. Zeng and H. Li, Sci. Rep. **7** (1), 1-9 (2017).

75. G. E. Hinton, Scholarpedia **4** (5), 5947 (2009).

76. H. Im, D. Pathania, P. J. McFarland, A. R. Sohani, I. Degani, M. Allen, B. Coble, A. Kilcoyne, S. Hong and L. Rohrer, Nature biomedical engineering **2** (9), 666-674 (2018).

77. X. Li, J. Tian and W. Shen, ACS Appl. Mater. Interfaces **2** (1), 1-6 (2010).

78. R. Safavieh, G. Z. Zhou and D. Juncker, Lab Chip **11** (15), 2618-2624 (2011).

79. M. Reches, K. A. Mirica, R. Dasgupta, M. D. Dickey, M. J. Butte and G. M. Whitesides, ACS Appl. Mater. Interfaces **2** (6), 1722-1728 (2010).

80. P. Bhandari, T. Narahari and D. Dendukuri, Lab Chip **11** (15), 2493-2499 (2011).

81. A. Nilghaz, D. H. Wicaksono, D. Gustiono, F. A. A. Majid, E. Supriyanto and M. R. A. Kadir, Lab Chip **12** (1), 209-218 (2012).

82. A. Nilghaz, D. R. Ballerini and W. Shen, Biomicrofluidics **7** (5), 051501 (2013).

83. G. Zhou, X. Mao and D. Juncker, Anal. Chem. **84** (18), 7736-7743 (2012).

84. Z. S. Ballard, H.-A. Joung, A. Goncharov, J. Liang, K. Nugroho, D. Di Carlo, O. B. Garner and A. Ozcan, NPJ digital medicine **3** (1), 1-8 (2020).

85. S. Rajaram, B. Pavie, N. E. F. Hac, S. J. Altschuler and L. F. Wu, Nat. Methods **9** (7), 634-635 (2012).

86. P. Mamoshina, A. Vieira, E. Putin and A. Zhavoronkov, Mol. Pharm. **13** (5), 1445-1454 (2016).

87. K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, Nature **559** (7715), 547-555 (2018).

88. P. Domingos, Commun. ACM **55** (10), 78–87 (2012).

89. D. de Ridder, J. de Ridder and M. J. T. Reinders, Brief. Bioinform. **14** (5), 633-647 (2013).

90. R. Miotto, F. Wang, S. Wang, X. Jiang and J. T. Dudley, Brief. Bioinform. **19** (6), 1236-1246 (2018).

91. T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. (Springer Science & Business Media, 2009).

92. A. Y. Ng and M. I. Jordan, presented at the Adv. Neural Inf. Process. Syst., 2002 (unpublished).

93. L. Ohno-Machado, J. Am. Med. Inf. Assoc. **18** (5), 539-539 (2011).

94. T. D. Gunter and N. P. Terry, J. Med. Internet Res. **7** (1), e3 (2005).

95. P. Kierkegaard, Computer law security review **27** (5), 503-515 (2011).

96. H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim and R. M. Summers, IEEE Trans. Med. Imaging **35** (5), 1170-1181 (2015).

97. G. Amit, R. Ben-Ari, O. Hadad, E. Monovich, N. Granot and S. Hashoul, presented at the Medical Imaging 2017: Computer-Aided Diagnosis, 2017 (unpublished).

98. K. S. Suslick, N. A. Rakow and A. Sen, Tetrahedron **60** (49), 11133-11138 (2004).

99. P. J. Mazzone, X.-F. Wang, Y. Xu, T. Mekhail, M. C. Beukemann, J. Na, J. W. Kemling, K. S. Suslick and M. Sasidhar, J. Thorac. Oncol. **7** (1), 137-142 (2012).

100. X. Zhong, D. Huo, H. Fa, X. Luo, Y. Wang, Y. Zhao and C. Hou, Sensors Actuators B: Chemical **274**, 464-471 (2018).

101. H. Kim, O. Awofeso, S. Choi, Y. Jung and E. J. A. O. Bae, **56** (1), 84-92 (2017).

102. D. Akkaynak, T. Treibitz, B. Xiao, U. A. Gürkan, J. J. Allen, U. Demirci and R. T. Hanlon, JOSA A **31** (2), 312-321 (2014).

103. P. L. Vora, J. E. Farrell, J. D. Tietz and D. H. Brainard, (1997).

104. Y. Zhang, Y. Wu, Y. Zhang and A. Ozcan, Sci. Rep. **6**, 27811 (2016).

105. B. Kurian and R. Liyanapathirana, presented at the Proceedings of the 13th International Conference on Damage Assessment of Structures, 2020 (unpublished).

106. F.-G. Yuan, S. A. Zargar, Q. Chen and S. Wang, presented at the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020, 2020 (unpublished).

107. E. S. Gargus, H. B. Rogers, K. E. McKinnon, M. E. Edmonds and T. K. Woodruff, Nature Biomedical Engineering **4** (4), 381-393 (2020).

108. S. N. Bhatia and D. E. Ingber, Nat. Biotechnol. **32** (8), 760-772 (2014).

109. C. D. Edington, W. L. K. Chen, E. Geishecker, T. Kassis, L. R. Soenksen, B. M. Bhushan, D. Freake, J. Kirschner, C. Maass, N. Tsamandouras, J. Valdez, C. D. Cook, T. Parent, S. Snyder, J. Yu, E. Suter, M. Shockley, J. Velazquez, J. J. Velazquez, L. Stockdale, J. P. Papps, I. Lee, N. Vann, M. Gamboa, M. E. LaBarge, Z. Zhong, X. Wang, L. A. Boyer, D. A. Lauffenburger, R. L. Carrier, C. Communal, S. R. Tannenbaum, C. L. Stokes, D. J. Hughes, G. Rohatgi, D. L. Trumper, M. Cirit and L. G. Griffith, Sci. Rep. **8** (1), 4530 (2018).

110. M. M. Gong and D. Sinton, Chem. Rev. **117** (12), 8447-8480 (2017).

111. C. M. Pandey, S. Augustine, S. Kumar, S. Kumar, S. Nara, S. Srivastava and B. D. Malhotra, Biotechnol. J. **13** (1), 1700047 (2018).

112. K. Pardee, A. A. Green, M. K. Takahashi, D. Braff, G. Lambert, J. W. Lee, T. Ferrante, D. Ma, N. Donghia and M. Fan, Cell **165** (5), 1255-1266 (2016).

113. R. Cavicchioli, W. J. Ripple, K. N. Timmis, F. Azam, L. R. Bakken, M. Baylis, M. J. Behrenfeld, A. Boetius, P. W. Boyd, A. T. Classen, T. W. Crowther, R. Danovaro, C. M. Foreman, J. Huisman, D. A. Hutchins, J. K. Jansson, D. M. Karl, B. Koskella, D. B. Mark Welch, J. B. H. Martiny, M. A. Moran, V. J. Orphan, D. S. Reay, J. V. Remais, V. I. Rich, B. K. Singh, L. Y. Stein, F. J. Stewart, M. B. Sullivan, M. J. H. van Oppen, S. C. Weaver, E. A. Webb and N. S. Webster, Nat. Rev. Microbiol. **17** (9), 569-586 (2019).

114. B. S. Lambert, J.-B. Raina, V. I. Fernandez, C. Rinke, N. Siboni, F. Rubino, P. Hugenholtz, G. W. Tyson, J. R. Seymour and R. Stocker, Nature Microbiology **2** (10), 1344-1349 (2017).

115. B. Neumann, T. Walter, J.-K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel, C. Cetin, F. Sieckmann, G. Pau, R. Kabbe, A. Wünsche, V. Satagopam, M. H. A. Schmitz, C. Chapuis, D. W. Gerlich, R. Schneider, R. Eils, W. Huber, J.-M. Peters, A. A. Hyman, R. Durbin, R. Pepperkok and J. Ellenberg, Nature **464** (7289), 721-727 (2010).

116. M. Held, M. H. A. Schmitz, B. Fischer, T. Walter, B. Neumann, M. H. Olma, M. Peter, J. Ellenberg and D. W. Gerlich, Nat. Methods **7** (9), 747-754 (2010).

117. J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon and S. T. C. Wong, J. Biomol. Screen. **13** (1), 29-39 (2008).

118. A. Akay and H. Hess, IEEE journal of biomedical and health informatics **23** (3), 906-920 (2019).