



Machine-Learning-Assisted Intelligent Imaging Flow Cytometry: A Review

Shaobo Luo, Yuzhi Shi, Lip Ket Chin, Paul Edward Hutchinson, Yi Zhang, Giovanni Chierchia, Hugues Talbot, Xudong Jiang, Tarik Bourouina, Ai-Qun Liu

► To cite this version:

Shaobo Luo, Yuzhi Shi, Lip Ket Chin, Paul Edward Hutchinson, Yi Zhang, et al.. Machine-Learning-Assisted Intelligent Imaging Flow Cytometry: A Review. *Advanced Intelligent Systems*, 2021, 3 (11), pp.2100073. 10.1002/aisy.202100073 . hal-03540960

HAL Id: hal-03540960

<https://hal.science/hal-03540960>

Submitted on 24 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Machine-Learning-Assisted Intelligent Imaging Flow Cytometry: A Review

Shaobo Luo, Yuzhi Shi,* Lip Ket Chin,* Paul Edward Hutchinson, Yi Zhang, Giovanni Chierchia, Hugues Talbot, Xudong Jiang, Tarik Bourouina,* and Ai-Qun Liu*

Imaging flow cytometry has been widely adopted in numerous applications such as optical sensing, environmental monitoring, clinical diagnostics, and precision agriculture. The system, with the assistance of machine learning, shows unprecedented advantages in automated image analysis, thus enabling high-throughput measurement, identification, and sorting of biological entities. Recently, with the burgeoning developments of machine learning algorithms, deep learning has taken over most of data analysis and promised tremendous performance in intelligent imaging flow cytometry. Herein, an overview of the basic knowledge of intelligent imaging flow cytometry, the evolution of machine learning and the typical applications, and how machine learning can be applied to assist intelligent imaging flow cytometry is provided. Perspectives of emerging machine learning algorithms in implementing future intelligent imaging flow cytometry are also discussed.

1. Introduction

Imaging flow cytometry is an analytical tool extensively used to detect, sort, and count phytoplankton, cells, and other microparticles.^[1–5] By combining high-throughput flow cytometry with various imaging acquisition technologies such as multispectral imaging,^[6] imaging flow cytometry is capable of capturing thousands, even millions of images with multiparametric morphology information, allowing automated high-throughput data collection. However, human experts are often required for performing image analysis on traditional imaging flow cytometry.

Intelligent imaging flow cytometry (IIFC), as shown in **Figure 1**, which combines imaging flow cytometry and artificial intelligence, has been demonstrated for imaging-based high-throughput biosensing.^[7–18] Artificial intelligence (particularly

Dr. S. Luo, Prof. G. Chierchia, Prof. T. Bourouina
ESYCOM, CNRS UMR 9007
Universite Gustave Eiffel
Noisy-le-Grand, Paris 93162, France
E-mail: tarik.bourouina@esiee.fr

Dr. S. Luo
Shanghai Gene Sense Biotech Co., Ltd
111 Xiangke Road, Zhangjiang High-Technology Park
Pudong New District, Shanghai 201210
China

Dr. S. Luo
School of Microelectronics
Southern University of Science and Technology
1088 Xueyuan Avenue, Nanshan District, Shenzhen
Guangdong 518055, China

Prof. Y. Shi
National Key Laboratory of Science and Technology on Micro/Nano
Fabrication
Department of Micro/Nano Electronics
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240, China
E-mail: yuzhi.shi@sjtu.edu.cn


Prof. Y. Shi, Dr. L. K. Chin, Dr. X. Jiang, Dr. A.-Q. Liu
School of Electrical & Electronic Engineering
Nanyang Technological University
50 Nanyang Ave, Singapore 639798, Singapore
E-mail: LKCHIN@mgh.harvard.edu; eaqliu@ntu.edu.sg

Dr. L. K. Chin
Center for Systems Biology, Harvard University
Massachusetts General Hospital
Boston, MA 02114, USA

Dr. P. E. Hutchinson
Life Sciences Institute
National University of Singapore
#05-02, 28 Medical Drive, Singapore 117456, Singapore

Prof. Y. Zhang
School of Mechanical & Aerospace Engineering
Nanyang Technological University
50 Nanyang Ave, Block N3, Nanyang Ave, Singapore 639798, Singapore

Prof. H. Talbot
Centre de Vision Numerique
Universite Paris-Saclay
CentraleSupélec, Saint-Aubin, Paris 91190, France

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202100073>.

© 2021 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202100073

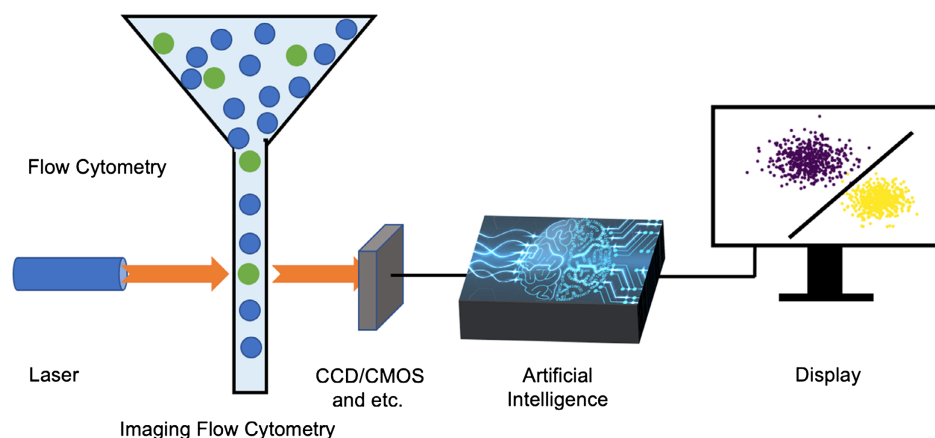


Figure 1. Overview of intelligent imaging flow cytometry.

deep learning) plays a critical role in IIFC by providing new approaches of image enhancement, reconstruction, correction, and more importantly automated object recognition and identification of cells and other targets of interest. Advances in artificial intelligence lead the development of IIFC. Several instances of IIFC using deep-learning models, such as VGGNet, GoogleNet, ZooplanktoNet, DenseNets, and deep active learning, have been demonstrated.^[19–22] A typical IIFC is shown in Figure 1, which combines flow cytometry, image acquisition technologies (laser/image sensors), and artificial intelligence. The system supports multiparametric analysis and high-throughput detection of the properties of a single cell from hundreds to millions of cells per second. IIFC is widely used in clinical diagnostics,^[23] environmental monitoring,^[24] and other potential biosensing applications.^[3,25–28]

In this Review, we focus on the recent developments in IIFC from the perspective of imaging technologies, the evolution of machine learning for computer vision, and machine learning techniques that have been developed specifically for IIFC. The emergent imaging technologies such as multispectral imaging,^[6] multi-field-of-view imaging,^[29] and serial time-encoded amplified microscopy (STEAM)^[30,31] are discussed to reveal more distinctive features of images. To understand cytometry imaging, we introduce the fundamentals of visual understanding and the evolution and knowledge of deep learning, which will increase the understanding of machine learning in a visual perception. Next, we review the interesting applications of machine learning in this field. Finally, we summarize the Review and give the perspectives of future development of machine-learning-assisted IIFC.

2. Imaging Technologies for Flow Cytometry

Technologies to obtain images with both high temporal and high spatial resolution are critical but challenging.^[6] The fundamental trade-off in imaging technologies is sensitivity, acquisition speed, and the amount of acquired information. There are two typical sensors used for imaging: 1) multipixelated imaging devices (camera-based), such as the charge-coupled device (CCD) and complementary metal–oxide–semiconductor (CMOS),^[32]

and 2) single-pixel photodetectors, e.g., the photomultiplier tube (PMT) and avalanche photodiode (APD).^[33]

The camera-based imaging flow cytometry has a dense 2D array of CCD or CMOS sensors, such as the commercial systems ImageStream (Figure 2a) and FlowSight, both developed by Millipore.^[34] They support multispectral imaging acquisition up to 12 images per cell and three different imaging modes—bright-field, scattering, and fluorescence—based on the technique time delay and integration (TDI).^[35–37] The TDI sensor includes multiple rows of CCD or CMOS sensors. When applied to imaging, the detecting objectives move along the column direction and the imaging data are shifted row by row. The system can read out a weak imaging signal without motion blur even with increasing exposure time. Unfortunately, data transfer between rows without gain (e.g., electron multiplication) also restricts the system to the limit of 3000 cells per second.

To increase the throughput, multi-field-of-view imaging flow cytometry^[29] was developed, as shown in Figure 2b. This method projects multiple fields of view into a 2D camera, such as microfabricating several microfluidic channels with $N \times M$ microlens arrays to capture multiple images simultaneously. Motion blur is a big problem in this kind of imaging cytometry when the targets move too fast and cannot be resolved by the imaging sensor under a fixed exposure time. Temporal coded excitation^[38] is a technique used to avoid motion blur, which uses a pseudo-random-code-modulated excitation pulse to illuminate the object.

PMT sensors^[33] provide superb sensitivity for photon signals with a high dynamic range, high bandwidth, and low dark noise, which serve as perfect candidates to implement high-throughput imaging flow cytometry. Normally, a laser scanner is used to generate the images from the time-domain signals collected from PMTs such as STEAM,^[30,31] as shown in Figure 2c. STEAM uses a near-infrared laser light with a wide spectral bandwidth as the illumination. The broadband laser pulses are encoded to 2D with two diffraction gratings for scanning and illuminating the cell. Eventually, the rainbow signal is collected by an APD detector. STEAM can achieve a throughput of 100 000 cells per second. Other examples using PMTs include fluorescence imaging by radiofrequency-tagged emission^[39,40] for high-speed fluorescence imaging, spatial-temporal transformation cytometry, etc.

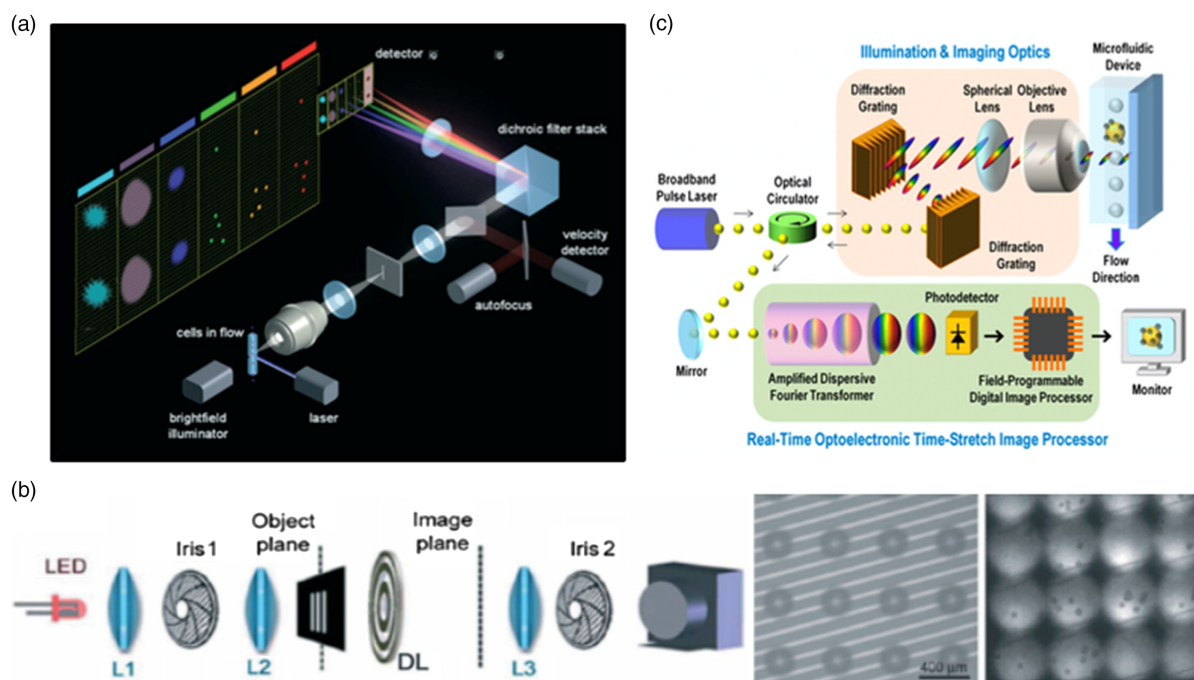


Figure 2. Optical systems of typical imaging flow cytometry. a) Optical configuration of ImageStream imaging flow cytometry. b) Multiple field-of-view imaging flow cytometer. c) Schematic illustration of STEAM flow analyzer. Reproduced with permission.^[6] Copyright 2016, Royal Society of Chemistry.

The emerging commercial imaging flow cytometry empowers high-speed cell sorting and microscopic imaging. For example, the ImageStreamX Mk II uses high-resolution and high-sensitivity objective lenses to capture bright-field, dark-field, and fluorescence images.^[37] The system contributes significantly to the advancement of a wide range of quantitative, statistically robust cellular analyses, cellular classification, cell-to-cell interactions, microalgae morphology, population dynamics, etc. FlowCam is another imaging flow cytometer that was originally developed by Fluid Imaging Technologies (Yarmouth, ME, USA) to study oceanic plankton.^[41] It uses a camera and flash illumination to snap the image of the moving particles in real-time. An image-processing software with machine learning algorithms is run to generate a single grayscale or color image of single cells. The software supports the extractions of different features such as area, area-based diameter, length, width, equivalent spherical diameter, and others properties.^[42] The Submersible Imaging FlowCytobot is another type of imaging flow cytometer that can be submerged in water up to 40 m depth for 6 months.^[43] It can also transmit acquired data to the cloud in real time. The Submersible Imaging FlowCytobot works similarly to a standard flow cytometer that uses hydrodynamic focusing to focus the sample stream and laser (e.g., 635 nm red diode laser for chlorophyll) to excite the particles for light scattering and fluorescence imaging, which allows us to analyze cells with size smaller than 150 μm .

3. Machine Learning

3.1. Machine Vision and Image Analysis

Imaging flow cytometry technologies enable capturing and analyzing images of cells with high quality and high throughput.

In addition to the challenges in image acquisition, storage, and processing, image analysis also requires significant efforts for the development of imaging flow cytometry, which promotes advances in machine vision.

The working principle of a machine vision system^[44] is elaborated here. First, an object is converted into an image signal through a machine vision device such as a camera. Then, the image signal is sent to a dedicated image-processing system to obtain the morphological information of the captured object. According to the pixel brightness, color, and spatial distribution, the imaging system performs various algorithms on those signals to extract the characteristics of the target object. Next, a control operation of the equipment is generated according to the result of the discrimination algorithms. The goal of computer vision is to fully understand the image of the electromagnetic wave from the reflection of the object surface, mainly the visible and infrared parts.

3.2. Traditional Machine Learning

Since 1960,^[45] a theoretical framework for object recognition has been conceptualized, as well as several general vision theoretical frameworks, visual integration theoretical frameworks, and many other new research methods and theories have emerged. Consequently, the processing of general 2D information and the research on the model and algorithm of 3D images have greatly improved and the machine has developed vigorously with emerging new concepts and theories. Before the invention of deep learning, the image analysis methods could be divided into the following five categories: image perception, image preprocessing, feature extraction, inference prediction, and recognition.^[46] In the early-stage development of machine learning, among the

dominant statistical machine learning groups, little attention was paid to features. The design principle of early-stage development of machine learning is to combine pixel values of the image in a statistical or nonstatistical form to express the part of or the whole object that one wants to identify or detect.

In 2001, a face-recognizing approach that was capable of working in real time using Haar-like features to locate a face was launched.^[47] Proposed in this approach, the Viola/Jones facial detector is a powerful binary classifier consisting of several simple classifiers, and is still widely used today. However, at the inception of the Viola/Jones facial detector, it was considered relatively time-consuming in the learning phase because adaptive boosting (Adaboost) is used to train the cascade of simple classifiers, such as finding the object of interest (e.g., a face). The model needs to split the input image into multiple rectangular blocks and then submit them to the cascaded weak detectors. If the patch passes through all stages of the cascaded weak detectors, it is classified as a positive example. Otherwise, the algorithm will reject the patch immediately. This whole process is repeated multiple times on various hierarchies of image scales.

In 2009, another important feature-based milestone work called deformable part models (DPM) as shown in **Figure 3** was developed.^[48] The DPM decomposes the object into partial subobjects, which follows the idea on the image model introduced in the 1970s, enforces a set of geometric constraints among them, and treats the simulated potential object center as a potential variable. The DPM excels at object detection tasks (using bounding boxes for localizing objects) and defeating template matching as compared to other object detection methods that were popular at that time whereby the histogram of oriented gradient (HoG)^[49] feature, as shown in **Figure 4**, was used to generate the corresponding “filter” for various objects. The HoG filter can record the edge and contour information of the object and use it as a filter at various positions in different pictures. When the output response value exceeds a certain threshold, the filter and the object in the picture are treated as highly matched, thus completing the detection of the object.

The HoG is a good feature descriptor that has been successfully deployed in human face detection problems.^[50] The HoG has an advantage on capturing the dense gradient information

of images, which is similar to scale invariant feature transform (SIFT)^[51] but the HoG demands fewer computation resources. The HoG is also resistant to the lighting conditions; e.g., it reduces shadows’ influence and other illumination variations such as smaller rotation and translation of the particle objects with the gradient and histogram algorithms. As shown in **Figure 4**, the HoG calculates on small blocks in a window of 8×8 pixels. In that window, the direction of gradient θ and the magnitude G are calculated by

$$G = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan \frac{G_x}{G_y} \quad (1a)$$

$$G_x = \mathbf{H}_x * I(x, y), \quad G_y = \mathbf{H}_y * I(x, y) \quad (1b)$$

where $I(x, y)$ is the input, \mathbf{H}_x is the vector $[-1 \ 0 \ 1]$, and \mathbf{H}_y is the vector $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$.

Finally, a gradient histogram of those 8×8 blocks is generated and put into nine bins. Each bin corresponds to the angles of direction of the gradient in $0^\circ, 20^\circ, 40^\circ, 60^\circ, 80^\circ, 100^\circ, 120^\circ, 140^\circ$, and 160° . As the gradient and magnitude of the image are mostly sensitive to the lighting, normalization on the histogram is desirable. The gradient histogram would result in more robust feature sets because it can eliminate the effect of variations when the lighting conditions are varying.

Local binary pattern (LBP) is a popular texture feature extraction method with an excellent performance in face detection.^[52,53] LBP excels in differentiating bright pixels from a dark background, which is used to describe edges, lines, spots, etc. The procedure of LBP feature extraction is shown in **Figure 5a**. First, the original input image is arranged into individual small cells with 8×8 pixels. Then, the LBP feature of each cell is calculated by comparing the intensity of the eight neighboring pixels with that center pixel and generating an 8 bit binary number in which 0 or 1 indicates that the intensity of the neighboring pixel is lesser or higher than the center pixel, respectively, as shown in **Figure 5b**.

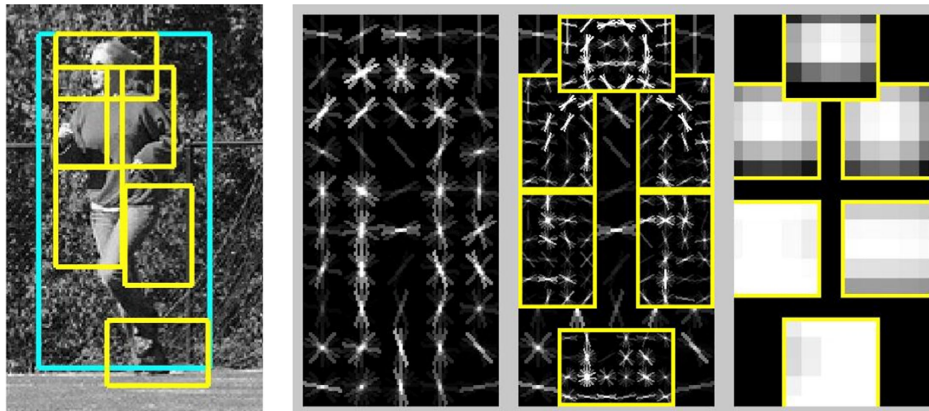


Figure 3. Detections obtained with a single-component person model. Example detection obtained with the person model. The model is defined by a coarse template, several higher-resolution part templates, and a spatial model for the location of each part. Reproduced with permission.^[48] Copyright 2008, IEEE.

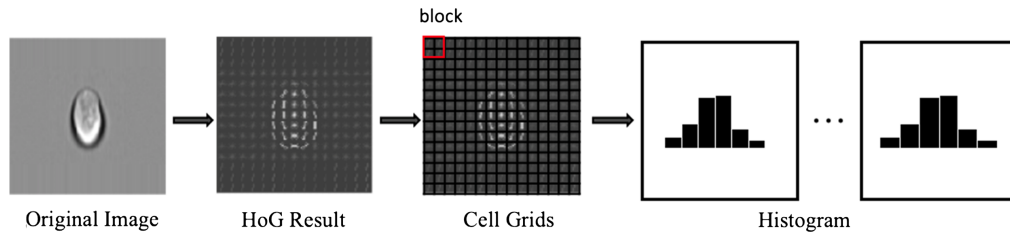


Figure 4. Working principle of histograms of oriented gradients.

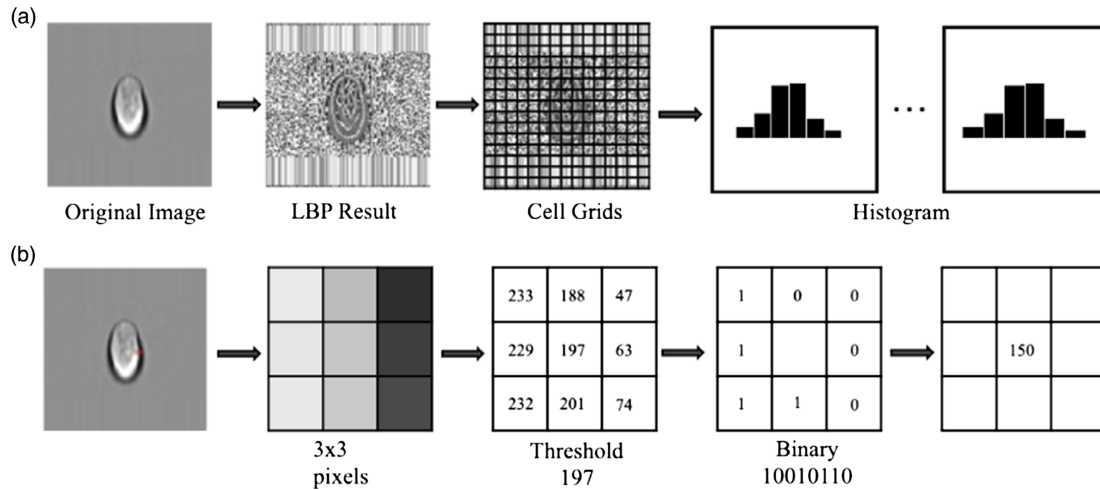


Figure 5. Local binary patterns. a) The procedure on local binary patterns histograms. b) The process steps on how to calculate local binary patterns.

Demonstrated examples of differentiating between Jurkat cells and white blood cells using traditional machine learning with imaging flow cytometry include the imaging flow cytometry data analysis that uses the features generated from CellProfiler and compares with the gradient boosting (GB)^[54] classifier or a random forest (RF)^[55] classifier to recognize the Jurkat cells.^[56] Another example, such as identifying the label-free white blood cells using the features generated from CellProfiler and comparing with five common classifiers such as K-nearest neighbors (KNNs), AdaBoost, GB, RF, or a support vector machine (SVM).^[57] The SVM classifier^[58] is one of the most popular discriminative classifiers before the era of deep learning, which translates the vector of training data into a high-dimensional space and performs the discrimination. By doing this, the optimal hyperplane can be generated, which splits the dataset into different classes via a training process. SVMs can be expressed as the following optimization problem

$$\begin{aligned} & \underset{W \in H, b \in R, \xi_i \in R}{\text{minimize}} && \begin{cases} \frac{1}{2} W^T W + C \sum_{i=1}^n \xi_i \\ \gamma_i (W^T \varphi(x_i) + b) > 1 - \xi_i \\ \xi_i \geq 0, \quad i = 1, \dots, n \end{cases} \\ & \text{subject to} && \end{aligned} \quad (2)$$

where the two-class problem (binary problem) was defined as $\gamma \in \{1, -1\}$. W is the weight, ξ is the margin constant, b is the bias, and $C \in \mathbb{R}_+$ is the regularization constant. The φ function

optionally projects the vector of training data into a high-dimension feature space \mathcal{H} by the so-called kernel trick, where the SVM can generate the boundary of decision surfaces easily. A good choice for φ is to use the radial basis function kernel as $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ and $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$ for the kernels.

A distance-based classifier such as the Mahalanobis distance classifier is an extension of the least-squares multiclass maximum likelihood classifier taking cross-correlations into account.^[59,60] The Mahalanobis distance classifier measures the number of standard deviation distance d with the calculated distance of x to a dataset and a mean u_i . The covariance matrix is defined as the equation \sum_i^{-1} and T is a standard transpose operation. The classification result is predicted by measuring the distance from x to classes i and assuming the result has the minimal distance from the cluster of true predicted class. The Mahalanobis distance can be reduced to the Euclidean distance when the covariance matrix is the identity matrix. The equation of the Mahalanobis distance is expressed as^[60]

$$d(x, u) = (x - u_i)^T \Sigma_i^{-1} (x - u_i) \quad (3)$$

Machine vision is used to determine whether a set of image data contains a specific object, image feature, or motion state. This problem can sometimes be solved automatically by an algorithm, but so far, there is no single method that can be widely used to perform well in varied situations, i.e., to identify any

object in unpredictable environments. The prior art can only solve well in the recognition of specific targets, such as simple geometric figure recognition,^[61] face recognition,^[62] printed or handwritten document recognition,^[63] and vehicle recognition.^[64] Unfortunately, the recognition often requires a specific lightening in a certain background, and designated target postures. Designing features by hand requires a lot of experience such as a profound understanding of the field. The algorithm may also require a lot of debugging. Moreover, machine vision engineers not only need to manually design features, but also need to design a more suitable classifier algorithm for the problem. The combination of designing features and choosing a classifier at the same time to achieve the best results is a difficult task, requiring well-trained experts.

3.3. Deep Learning

Machine vision systems are developed such that users do not need to manually design features and choose classifiers. It is desirable for machine vision systems to learn features and classifiers simultaneously, which means that when a user designs a certain model, the input is just a picture, and the output is its label. With the rapid development of deep learning, the emergence of convolutional neural networks (CNNs) has made this idea possible, and the research of computervision based on deep learning has also developed rapidly. LeCun proposed the first CNN in LeNet^[65] in 1998, as shown in Figure 6. The input image is a 32×32 grayscale image. The first layer undergoes a set of convolution sums and generates six 28×28 feature maps (C1), which pass a pooling layer to get six 14×14 feature maps (S2) and pass a convolution layer to generate sixteen 10×10 convolution layers (C3). Next, they pass the pooling layer to generate sixteen 5×5 feature maps (S4). It was used to classify handwritten digits 0–9 with two fully connected layers as the final layers. In 2012, a deeper and wider neural network AlexNet was published, which achieved a breakthrough with proposed 10% higher accuracy than traditional methods in ImageNet large scale visual recognition challenge (LSVRC).^[66] Nowadays, deep learning has been applied to a variety of areas and huge progress has been made in those fields, including visual recognition,^[67] speech recognition,^[68] biomedicine,^[69] and natural language processing.^[70]

Deep-learning methods are well suited to constructing architectures that can be trained end to end from image data to achieve cell classification. This approach reduces manual laboring in the traditional approach, as shown in Figure 7. It can automatically build multiple levels of representation of data with abstraction. For example, the first layer studies the edge or color information. The second layer studies the motif information. The third layer may learn the eyes and nose information. Finally, the deep-learning method can learn the weights for the classifier to detect the human face. The importance layers in the deep neural network are the convolutional layer, active layer, and pooling layer (Figure 8), e.g., the CONV layer (convolutional layer (Convolution) + the ReLU layer (Activation)), and the fully connected layer (FC layer).

The convolution function is used to extract the features from the input. The basic operation of convolution is shown in Figure 9a. On the left side of the figure, the input has a dimension of $32 \times 32 \times 3$. It convolutes with a kernel H with a size of $3 \times 3 \times 3$. Finally, a feature with $30 \times 30 \times 3$ dimensions is generated, which is calculated by sliding the kernel from the top-left corner to the bottom-right on the input line by line and one layer of output is generated by the operating of element by element multiplied and accumulated with the kernel. For example, ten kernels will generate ten layers of output.

The ReLU layer, as shown in Figure 9b, is a rectified linear unit activation function. It implements a nonlinear “trigger” function with the formula $y = \max(x, 0)$, while the input has the same size as the output layer. The ReLU layer outputs zero when the input is negative. Compared with other nonlinear functions such as a sigmoid, hyperbolic tangent, and absolute of hyperbolic tangent, the networks with ReLU learn severalfold faster than other nonlinear functions. The max-pooling layer as shown in Figure 9c is used to reduce the resolution of the features. It makes the features more robust with lower noise and distortion. For instance, the pooling layer cuts down the sample from the input dimension of $224 \times 224 \times 64$ into an output dimension of $112 \times 112 \times 64$ with a filter size of 2×2 and stride with two steps.

One or several fully connected layers are normally added to the last layer of a CNN and acts as a classifier for the final decision. The full connected layer always takes a vector of m input (X) as the input volume and generates n output (Y) with a function that is expressed as

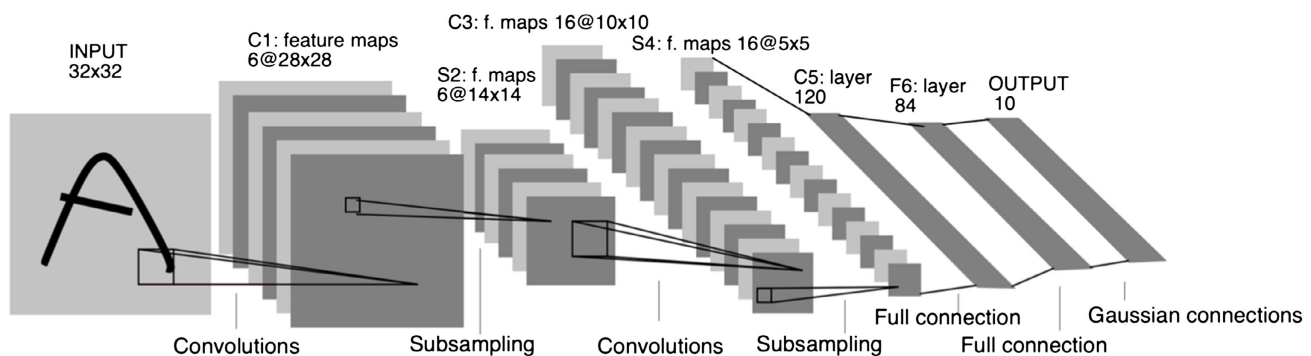


Figure 6. The architecture of the LeNet-5 neural network. A CNN for handwriting digital recognition. Reproduced with permission.^[65] Copyright 1998, IEEE.

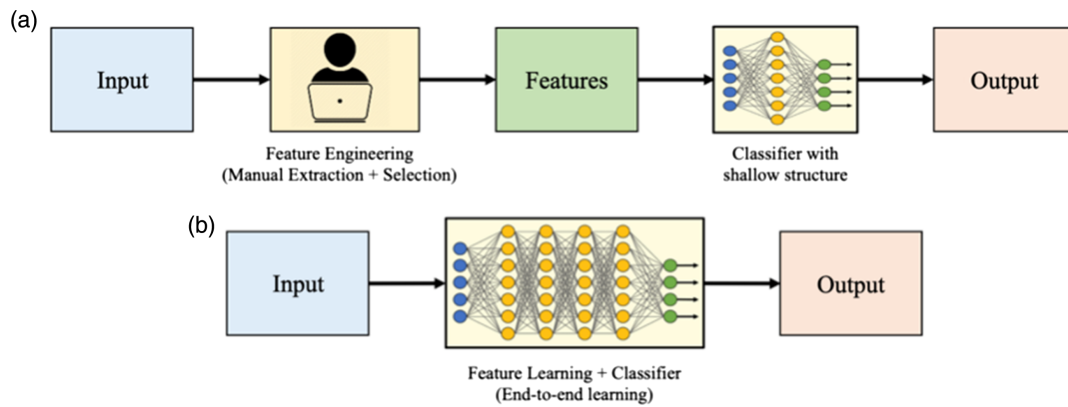


Figure 7. Comparison between a) traditional machine learning and b) deep learning for classification.

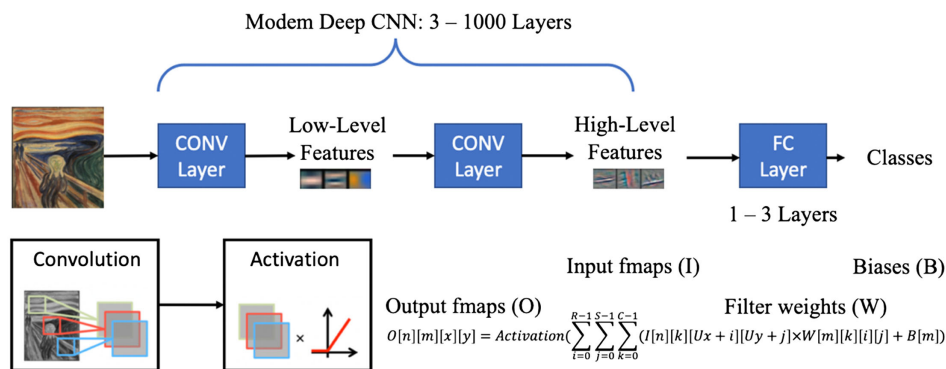


Figure 8. A convolutional neural network.

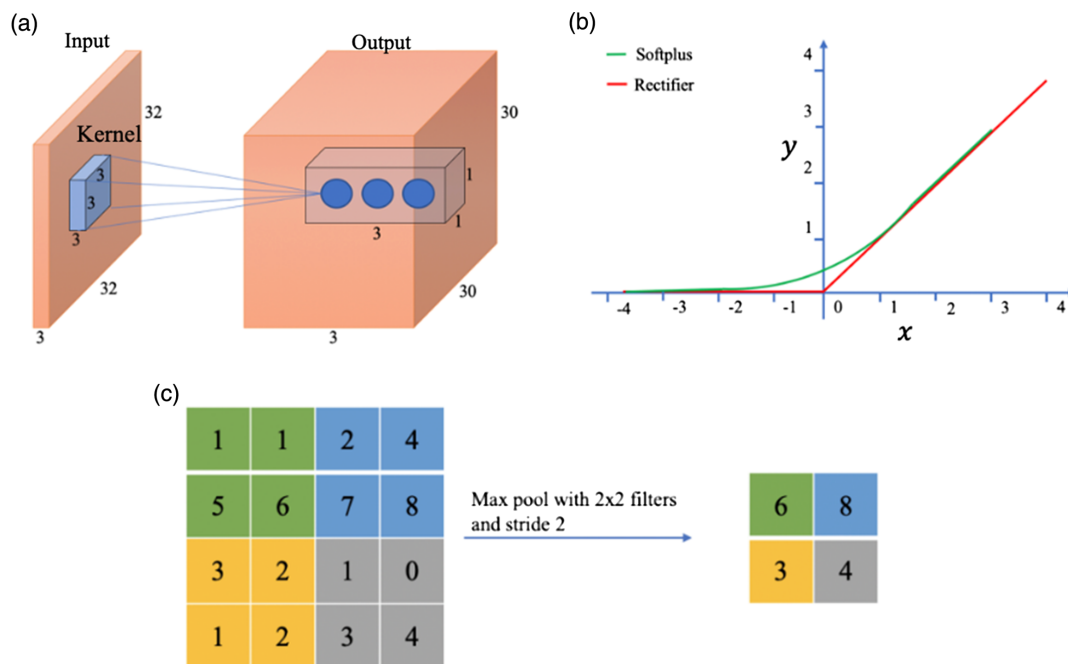


Figure 9. Layers of a CNN. a) Convolutional operation. b) Rectified linear unit (ReLU). c) Max-pooling operation.

$$Y_n = \mathbf{W}\mathbf{X}m + b \quad (4)$$

where m is the input dimension, which is computed with the weight matrix \mathbf{W} with matrix multiplication and added to a bias offset b .

The learning and optimization process is used to generate the optimal values of the trainable parameters such as kernel weights in convolutional layers and the weights in dense layers. The parameters are optimized by the backpropagation algorithm, which uses a gradient descent (GD)^[71] method to optimize the model iteratively by minimizing a loss function (e.g., cross-entropy loss). The three frequently used GD methods are batch gradient descent, stochastic gradient descent, and minibatch gradient descent. Softmax regression^[72] of the classification layer outputs was used to train the network, which can be written as

$$y_j = \frac{\exp\{x_j^k\}}{\sum_{i=1}^n \exp\{x_i^k\}}, j = 1, 2, \dots, n \quad (5)$$

where x_i^k is in the input and y_j is the output probability.

During the training, the loss is calculated from the model input with forwarding propagation whereby the loss difference backward propagates from the output to the input layer to generate the gradient of each layer. The parameters of every layer are updated with that gradient and the parameters of the model are converged after the iterative process.

3.4. Recent Advances in CNNs

A CNN is a powerful neural network that is widely used for image classification and segmentation. The CNN is inspired by the natural visual perception mechanism from the human perception system. The early attempt was the proposed neocognitron system in 1980.^[73] By improving the structure of the neocognitron, LeCun proposed LeNet-5 to solve handwritten digits, which established the modern framework of the CNN.^[65] LeNet-5 gave a basic idea of the CNN that uses a three-tier architecture: convolution, downsampling, and nonlinear activation functions. A CNN extracts image space features using convolution and reducing image average sparsity with downsampling. The activation function takes a hyperbolic tangent or sigmoid function. A multilayer neural network as the final classifier uses sparse connection matrices between layers to avoid large computational costs. LeNet-5 can be trained using the backpropagation algorithm and derive an effective representation of the original image, which allows the CNN to recognize the object directly from the original pixels with minimal preprocessing. However, due to the lack of large-scale training data and limited computing power, LeNet-5 could not work well on complex problems. From 1998 to 2010, the developments of neural networks was intense in the machine learning community, but it was not highly visible to the computer vision community. The rich dataset, the advanced theories in deep learning such as improving neural architectures, optimization methods (stochastic gradient descent, Nesterov accelerated descent,^[71] etc.), and the hardware improving (e.g., GPUs, low-power CPUs, fast- and low-latency disks such as single-shot detectors (SSDs)) have brought cost-effective hardware to the world, making deep neural network

computation affordable, and opening the door for deep learning. In 2010, a GPU neural network was published.^[74]

In 2012, AlexNet was published,^[66] which is relatively deeper than LeNet's network and won the first champion of the 2012 ImageNet Challenge,^[66] as shown in **Figure 10**. AlexNet not only has deeper neural networks, but also learns more complex features in the rich image dataset than LeNet. AlexNet introduced the ReLU function instead of tanh as its activation function, which is convex and has no vanishing gradient for positive weights, considerably reducing computation time in the learning phase. Furthermore, AlexNet used the dropout technique to clip certain neurons during training to avoid overfitting. It also introduced max-pooling technology and significantly reduced training time with a GPU. After the success of AlexNet, the researchers proposed other architectures, such as VGG,^[75] GoogleNet,^[76] residual network (ResNet),^[77] MobileNetV2,^[78] SENet,^[79] and BiT-L (another version of ResNet).^[80]

Regarding the structure, one of the CNN's development directions is focused on increasing the number of layers. As the ILSVRC 2015 champion, ResNet has 20 times more layers than AlexNet and 8 times more layers than VGGNet. By increasing the depth, the network can use the increased nonlinearity to derive the approximate structure of the objective function while yielding better performance than previous networks. However, this also increases the overall complexity of the network (more layers) and makes the network difficult to optimize and easily overfit. In addition, the optimization problem becomes more difficult when the network becomes deeper, with a larger parameter space. Therefore, simply increasing the depth of a network will result in higher training error. For example, the accuracy of a 56-layer network is not as good as that of the 20-layer network. In view of the effect from the layer, ResNet was designed with a residual module that allows us to train deeper networks.^[77]

The core idea of ResNet is to add a direct connection channel (X) to the network, known as identity shortcut connection.^[81] The network structure of traditional deep learning is a nonlinear transformation that is performed on the input, whereas ResNet allows the original input information to be passed directly to the subsequent layers, as shown in **Figure 11**. Traditional convolutional networks or fully connected networks will have more information loss during information transmission. Consequently, they will also cause gradients to disappear or explode and make deep networks unable to train. ResNet solves this problem to a certain extent as it protects the integrity of the information by directly bypassing the input information to the output. The entire network only needs to learn the part of the difference between input and output, simplifying the learning objectives and difficulty. A comparison of VGGNet (e.g., VGG-19) and ResNet is shown in **Figure 12**. The biggest difference between VGGNet and ResNet is the use of bypass connection to directly connect the input to the subsequent layers, which is also called shortcut or skip connections.

Various methods have been proposed to improve network performance in various aspects. The recent improvements of CNN include the convolutional layer, pooling layer, activation function, loss function, regularization, optimization, and fast computing techniques, for example, the inverted residual block (IRB), which was first introduced by the MobileNetV2^[78] architecture that includes a 1×1 expansion convolutional

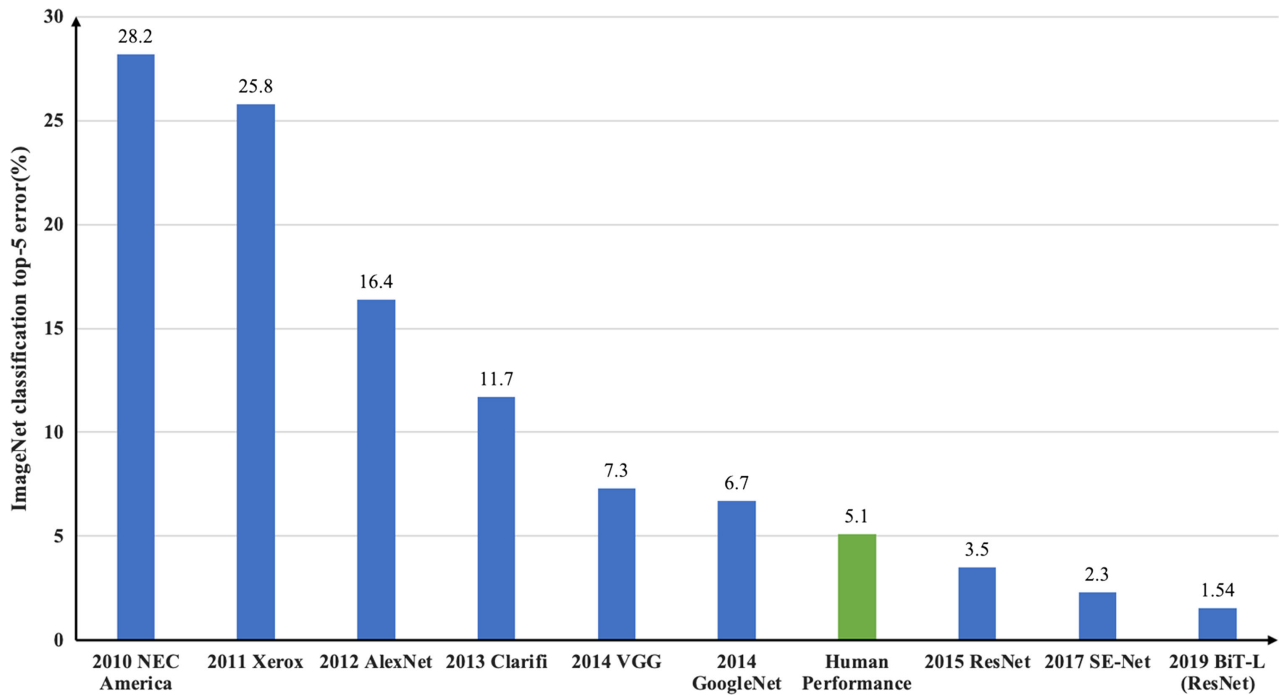


Figure 10. ImageNet challenge.

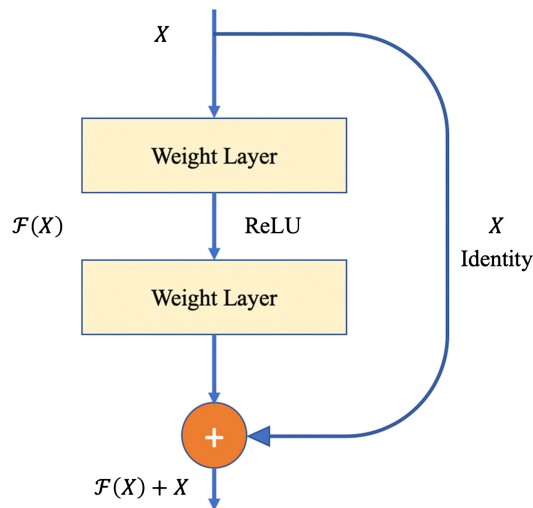


Figure 11. Residual learning building block. The core idea of ResNet is to add a direct connection channel to the network, known as identity shortcut connection. The network structure of traditional deep learning is a non-linear transformation that is performed on the input while ResNet allows the original input information to be passed directly to the subsequent layers.

layer, a depthwise convolution layer, and a 1×1 projection. The depthwise convolution layer and projection layer are referred to as the depthwise separable convolution adopted by Xception.^[82] The depthwise separable convolution^[83] splits the traditional convolution operation into two separated steps by two convolutions: the depthwise convolution and the pointwise convolution. The

depthwise convolution uses a separable filter with one filter per input channel to produce the output channel, as shown in Figure 13a. The depthwise convolution is represented as

$$\hat{\mathbf{X}}_{x,y,z}^k = \delta \left(\sum_{i,j} \hat{\mathbf{F}}_{i,j,z}^k \cdot \mathbf{X}_{x+i-1,y+j-1,z}^{k-1} + b_z^k \right) \quad (6)$$

where δ is an activation function, and b is a bias; $\hat{\mathbf{F}}^k$ is the depthwise filter in which the z th channel in $\hat{\mathbf{F}}^k$ only calculates with the z th channel of \mathbf{X}^{k-1} and produces the feature $\hat{\mathbf{X}}^k$ in the z th channel. A pointwise convolution uses a 1×1 filter to produce the final activation map as shown in Figure 13b. Compared to the traditional convolution, the computation saving of the depthwise separable convolution is $\frac{1}{N} + \frac{1}{D_k}$, where N is the number of output channels, and D_k is the kernel size. Furthermore, the IRB also increases the memory efficiency with its unique architecture. In addition, the skip connection structure is introduced to the IRB, which allows the network to access features in earlier stages and leads to a deeper neural network with high efficiency.

Metric learning is used to learn a distance function that measures similarity whereby similar targets are associated with a small distance, and dissimilar ones with a large distance.^[84] Deep metric learning (DML) currently mainly uses the deep-learning-based basement network to extract embedding, and then uses the L2 distance to measure the distance in the embedding space. In general, DML consists of three parts: a feature extraction network to map embedding, a sampling strategy to combine the samples in a minibatch into many subsets, and finally the loss function calculates the loss on each subset as shown in Figure 14. For example, in deep metric learning with

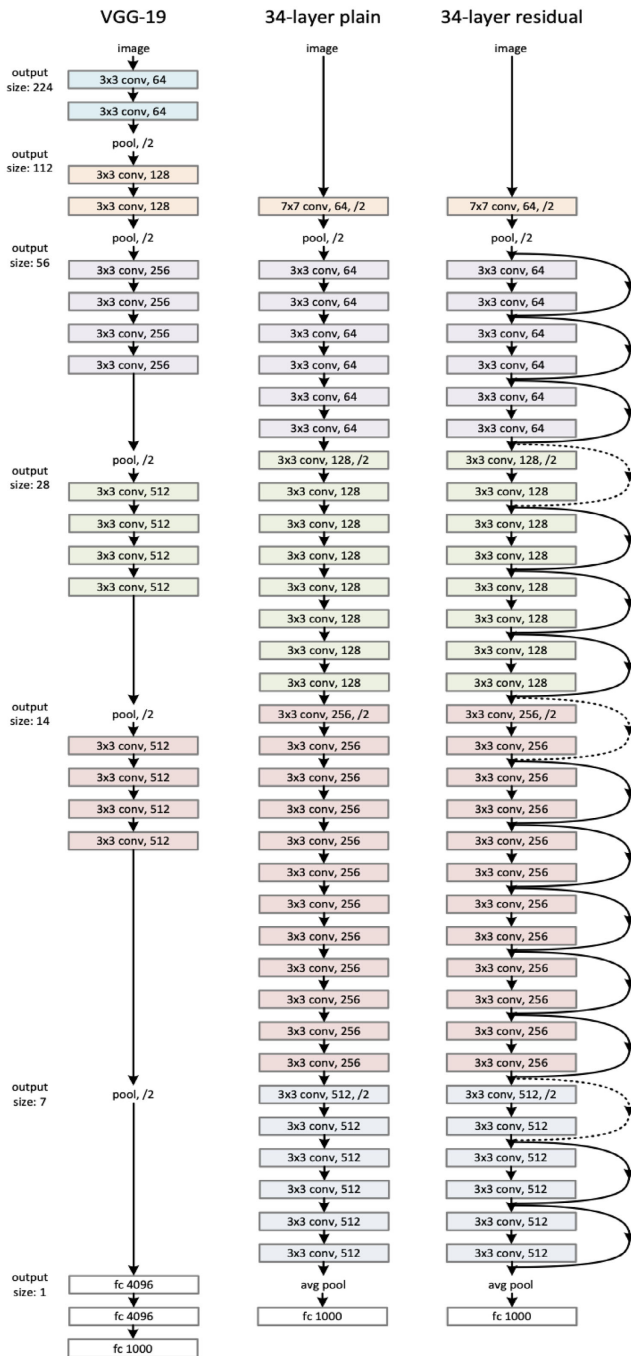


Figure 12. Examples of network architecture for ImageNet. The VGG-19 model is on the left, a plain network with 34 parameter layers is in the middle, and a residual network with 34 parameter layers is on the right. The dotted shortcuts increase dimensions. Reproduced with permission.^[77] Copyright 2016, IEEE.

contrastive loss^[85] the sampling algorithm randomly selects an example as an anchor and then randomly selects another image from the rest of the images of the same class as the positive pair and selects one image from the other classes as the negative pair. Popular loss functions are contrastive loss,^[85] triplet loss,^[86] etc.

The contrastive loss is used to train Siamese networks. For the pair of input $(\mathbf{x}_i, \mathbf{x}_j)$, it is a positive pair if \mathbf{x}_i and \mathbf{x}_j are semantically similar and a negative pair if they are dissimilar. The contrastive loss (L) is expressed as^[85]

$$L(\{W^{(m)}, b^{(m)}\}_{m=1}^M) = \sum_{(i,j) \in S} h(d_f(\mathbf{x}_i, \mathbf{x}_j) - \tau_1) + \sum_{(i,j) \in D} h(\tau_2 - d_f(\mathbf{x}_i, \mathbf{x}_j)) \quad (7)$$

where $h(x) = \max(0, x)$ is the hinge loss function, W is the weight, and τ_1 and τ_2 are two positive thresholds with $\tau_1 < \tau_2$. $S = \{(i, j)\}$ are the similar pairs and $D = \{(i, j)\}$ are the dissimilar pairs, and the Euclidean distance d_f between \mathbf{x} and \mathbf{y} is expressed as

$$d_f(\mathbf{x}, \mathbf{y}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|_2 = \sqrt{(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}))^T (\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}))} \quad (8)$$

where $\mathbf{x}, \mathbf{y} \in \chi$. The triplet loss is expressed as^[86]

$$L(x_a, x_p, x_n) = \max\left(0, m + \|\mathbf{f}(x_a) - \mathbf{f}(x_p)\|_2^2 - \|\mathbf{f}(x_a) - \mathbf{f}(x_n)\|_2^2\right) \quad (9)$$

where x_a and x_p are in same class and x_n is in a different class.

4. Deep Learning in Imaging Flow Cytometry

Intelligence imaging flow cytometry that combines imaging flow cytometry and artificial intelligence has emerged as a promising platform for imaging-based high-throughput biosensing.^[87,88] With the developments of cell-imaging techniques, the imaging system could reflect the rich sets of cell information that allow insightful and more rigorous analysis based on the fluorescence signal through immunolabeling and the scattering signal originated from the interaction of the cell structures with light. Based on the spatial arrangement of the cell images, the analysis problem can be split into two classes: 1) images that contain multiple cells and 2) images that contain only a single cell.

When one image contains multiple cells, detection and tracking problems occur. The detection task localizes all objects in the image with a bounding box such as when applying deep learning to detect mitosis. As to the evolution with deep learning, two-stage models such as Faster-RCNN are used to detect cells infected by malaria parasites^[89] and one-stage models such as SSD to detect neural cells.^[90] Compared with the previous models, for instance, RCNN, Faster-RCNN and SSD provide better performance on speed and detection accuracy. Nowadays, segmentation and detection joint approaches such as DeepLab^[91] and Mask-RCNN^[92] excel with the advantage of multitask learning.

For the detection and tracking task, a pipelined real-time imaging processing algorithm based on a CNN has been demonstrated for imaging flow cytometry.^[10] This algorithm uses a simplified CNN to identify microbeads and cells. A microfluidic channel is monitored by a CMOS camera via a microscope. Then, the real-time image is processed by a real-time moving object detector (R-MOD) system as shown in **Figure 15**. The

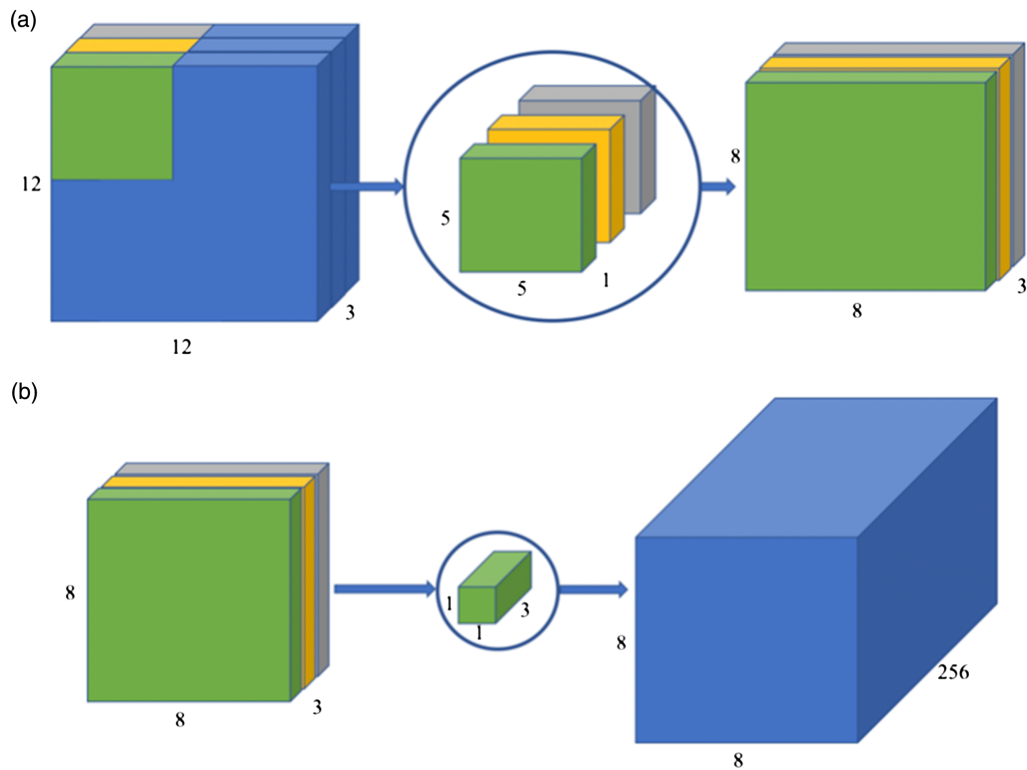


Figure 13. a) Depthwise convolution. b) Pointwise convolution with 256 kernels.

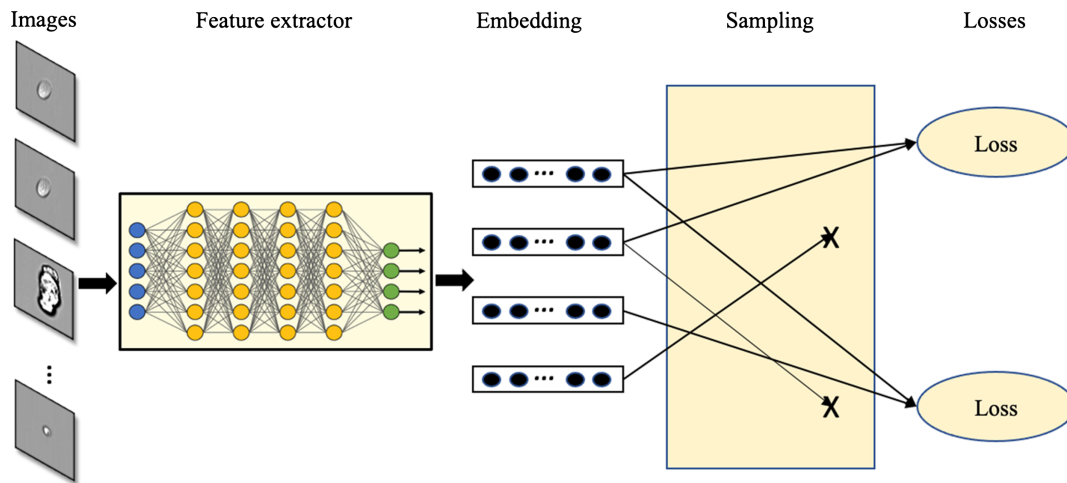


Figure 14. General pipeline of deep metric learning. It includes a base network for extracting features to the latent embedding space, sampling strategies, and loss functions.

R-MOD system contains two parts: 1) multiple-object tracking and 2) single-cell image acquisition and identification. The multiple-object-tracking algorithm is composed of three parts: image segmentation, detection, and tracking. The image segmentation algorithm is implied with a CNN, which performs a regression operation to convert the grayscale microscopy image to a probability density map. The detection algorithm finds the mean μ and variance σ of the cell object. First, it finds the

maximum pixel in the density map as the mean μ . Then, the area around the maximum pixel (p_{\max}) with σ is cropped from the density map. σ is expressed as

$$\sigma^2 = 0.5p_{\max} \quad (10)$$

This process continues until all Gaussian distributions in the density map have been removed. The tracking algorithm uses the

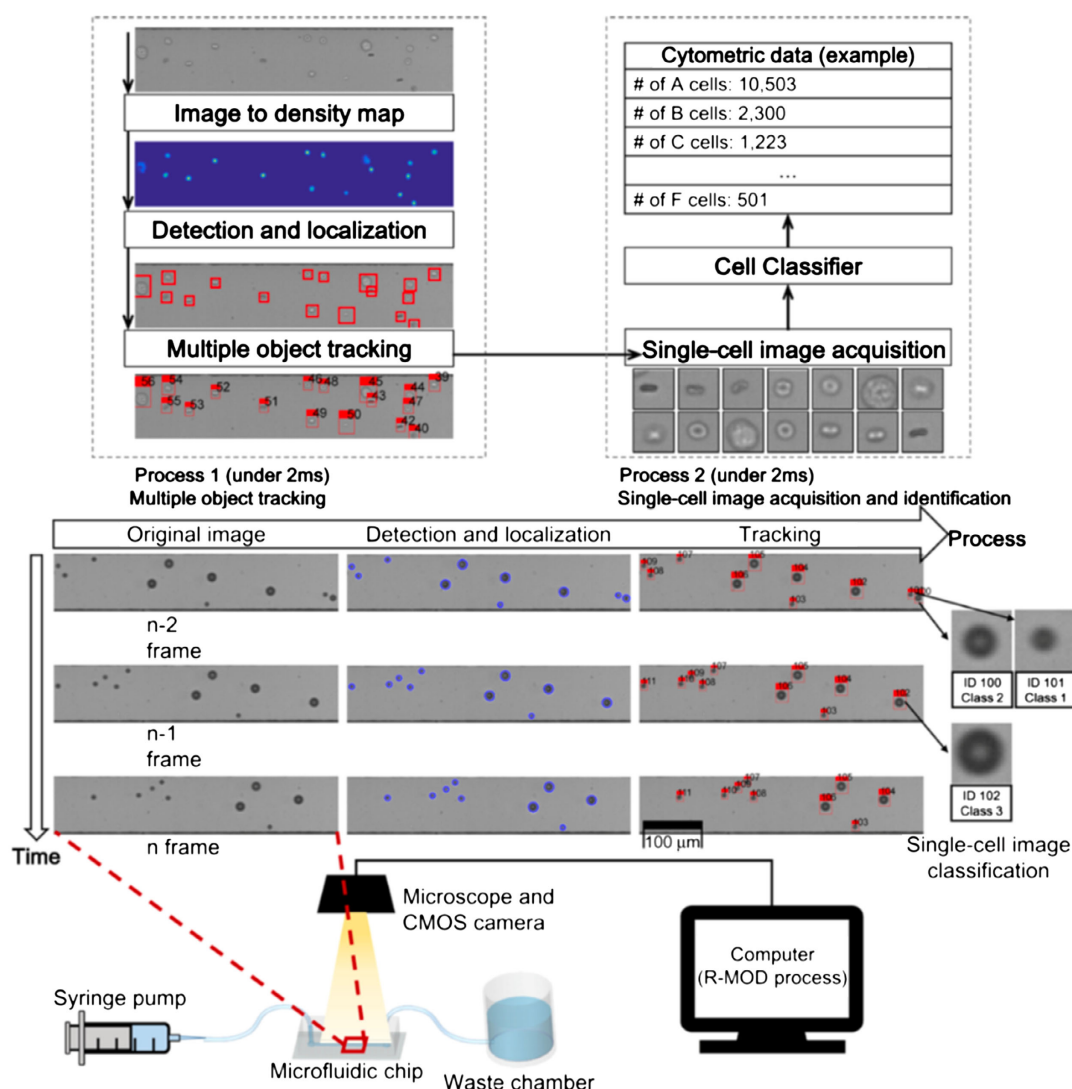


Figure 15. R-MOD (real-time moving object detector) system. Reproduced with permission.^[10] Copyright 2017, Springer Nature.

Hungarian algorithm to detect objects in the consecutive images and assigns an object number to the detected objects. About 21×21 pixels are cropped around the center of the object. A classifier-based CNN is used to determine the cell type without background noises from other undetermined objects. They also do not account of the natural containment particles in the water. Furthermore, the computation power of this classifier-based CNN is high and needs to be run on a high-end GPU workstation. Future considerations on object detection and tracking algorithms for imaging flow cytometry include recurrent YOLO,^[93] SiamMask,^[94] Deep SORT,^[95] and Tracking R-CNN.^[96]

Classification and segmentation are two fundamental problems in computer vision as well as single-cell image analysis. The image classification as in **Figure 16a** predicts a label (*Giardia*) for an input image, and the segmentation, as shown in **Figure 16b**, splits the digital imaging into subparts or superpixels such that every superpixel has the same meaningful label. In general, the segmentation is a subset of the classification,

which predicts in pixel level. In an image with multiple cells, the algorithms split the image of multiple cells into single cells or subcellular parts and predict the label of each single cell or subcellular part.

Early attempts used fully convolutional neural networks (FCNNs) on segmentation and classified cell such as the HepG2 cell specimen.^[97] The FCNN^[98] was a type of state-of-the-art architecture on the image segmentation task in 2014 and was trained to classify the HepG2 cell specimen into seven categories. Compared with the classification network, the FCNN replaces the fully connected layers to 1×1 convolutional layers. The classification network outputs a label for each image, but the FCNN gives a pixel label for every pixel. It learns a function that maps the input pixel to the output pixel label. The FCNN and CNN are different because the last three layers in the CNN network are 1D vectors. The calculation method no longer involves convolution. Therefore, the 2D information is lost. However, in FCNNs all three layers are converted into a 1×1

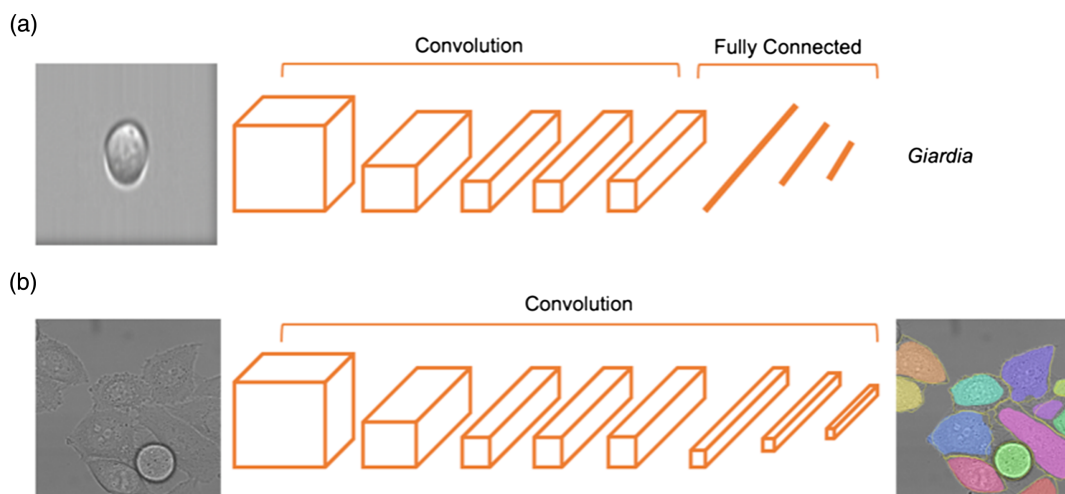


Figure 16. Classification and segmentation. a) Classify one image into a label; b) segment multiple cells into an individual.

convolution kernel with the equivalent vector length corresponding to the multichannel convolutional layer and the latter three layers in the FCNNs are all in convolution-based calculations. In the whole model, all layers are convolutional layers and without fully connected layers.

Recently, U-Net and its variations have dominated cell-level segmentation and counting.^[99,100] U-Net is a generic FCNN with concatenation at multiple scales. The U-Net architecture takes the features from multiple layers into account and provides good

localization with utility on the context for pixel-level classification. **Figure 17** shows the basic structure of the U-Net. The left part is the contracting path, which follows the standard approach of the traditional architecture of the CNN. In every block, it has two 3×3 unpadded convolutions with each followed by the ReLU layer. At the end of each block, a 2×2 max-pooling down-sampling layer is attached. The right side is the expansive path with consecutive blocks of 2×2 up-convolutional layer and 3×3 convolutional layers. To increase the local information,

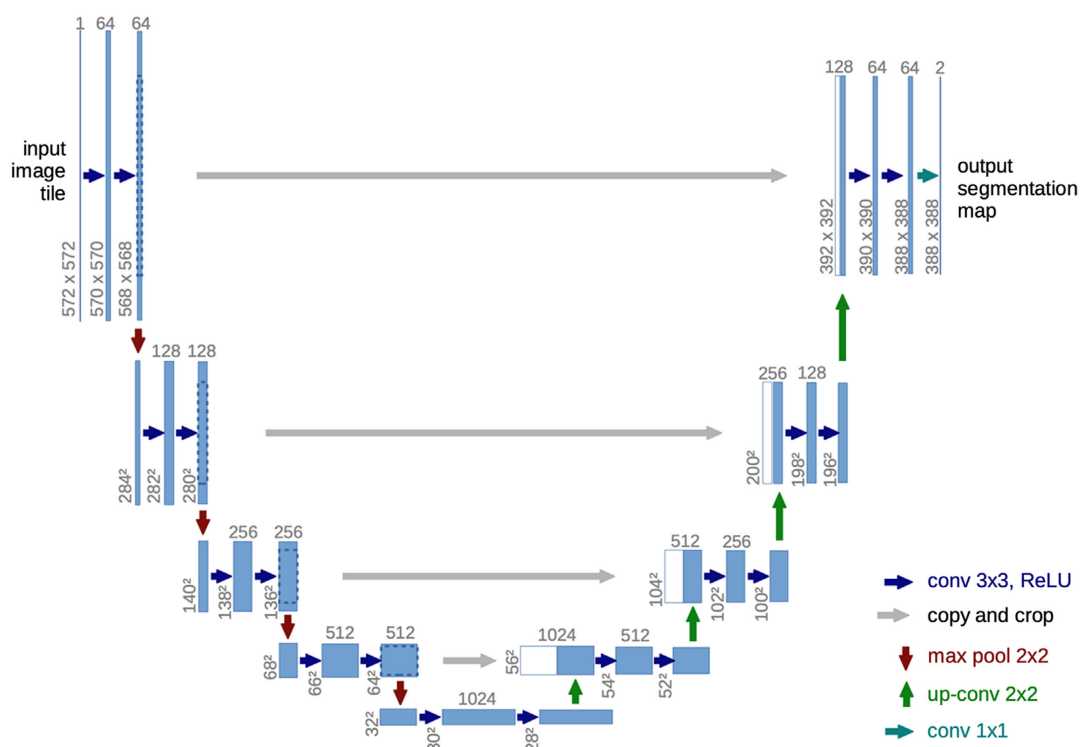


Figure 17. U-Net: the U-Net architecture takes the features from multiple layers into account and provides good localization with utility on the context for pixel-level classification. Reproduced with permission.^[99] Copyright 2015, Springer.

information from the contracting path is combined with concatenation. At the end of the whole network, a 1×1 convolutional layer is applied to map the feature map from 64 to 2 classes in the depth direction (cell and membrane).

Examples of using deep learning in bioparticle classification and detection on noncommercial IIFC and commercial IIFC are summarized in the Table 1 and 2, respectively. One of the examples is a deep CNN used as an early-warning system for anthrax detection.^[8] As shown in Figure 18, a dataset was built with different anthrax samples such as *Bacillus anthracis*, *Bacillus thuringiensis*, *Bacillus cereus*, *Bacillus atrophaeus*, and *Bacillus subtilis*. A deep CNN, HoloConvNet, was built with CNN for anthrax classification. It has three convolutional layers, two fully connected layers, and achieves a classification accuracy of 96.3%.

An ImageJ plugin interface to U-Net is shown in Figure 19 to enable users to count, segment, and detect cells.^[100] U-Net with the ImageJ interface offers a step-by-step protocol for cell detection, counting, and segmentation, such as prediction of

the center of the cell and delineation of the outlines of individual cells. U-Net can achieve results comparable with the level of human experts. For training, it requires a relatively low number of annotated images with special data augmentation, which needs more data for training on particularly difficult cases.

A deep CNN-based real-time cell sorter^[22,101,102] with a processing speed of 2000 events per second was also demonstrated, as shown in Figure 20. However, the system requires a complex hybrid hardware/soft data management system 8 GPUs (NVIDIA GeForce GTX 1080 Ti) for image processing. First, the suspended cells in a sample tube are injected into an intelligent IACS, which are hydrodynamically focused to a single stream. Then, the cells are imaged by virtual-freezing fluorescence imaging.^[103] Next, the images are analyzed by a real-time intelligent image processor. Finally, the cells are sorted by a dual membrane, which receives decisions from the image processor and uses a cell push-pull mechanism for sorting. The whole process is operated in an automated and real-time manner.

Table 1. Summary on machine learning articles by new imaging flow cytometry methods.

Application	Imaging	Application	Label	Algorithm	Result	Hardware	Reference
Cell classification	Time-stretch imaging	Classify the cells. THP1, MCF7, MB231, PBMC	Label-free	CNN model	99% accuracy	Nvidia Tesla K40c	[112]
Cell sorting		Cell sorting	Label-free	CNN model	NA	Nvidia GTX-1080Ti	[22]
Natural water samples analysis	Lens-free holographic imaging	Phase-contrast color image reconstruction and identification of plankton	Label-free	CNN model	NA	NA	[13]
Bioaerosol sensing	Frequency-division-multiplexed (FDM) microscope	Label-free bioaerosol detection	Label-free	Resnet based	Accuracy >94%	NA	[11]
Real-time inference and cell sorting	Time-stretch imaging	Cell sorting and cancer cell classification		CNN model	95% accuracy	Nvidia Tesla P100 GPU and Nvidia K80 GPU	[15]
Image construction and classification	Lens-less time-resolved holographic speckle imaging	–	Label-free	CNN model	–	Nvidia GeForce GTX 1080Ti GPU	[21]
Detection of cellular drug responses	Time-stretch imaging	Drug-treated and -untreated cell classification	Label-free	Linear SVM/CNN model	Accuracy of 92%	NA	[9]
Cell cycle analysis	Microscopy	Label-free cell-cycle classification of Jurkat cells	Label-free	LSBoosting	70.2 ± 2.2% (G1), 90.1 ± 1.1% (S), 96.8 ± 0.3% (G2), and 44.0 ± 8.4% (M)	NA	[7]
Parasite detection	Lens-less time-resolved holographic speckle imaging	Parasitic detection	Label-free	CNN model	68–76% (at the lower end of our tested concentrations) to ≈38–39% (at the higher end of our tested concentrations)	Nvidia GTX 1080	[113]
Detection	Microscopy	Detection beads	Label-free	CNN model	93.3% mAP	Nvidia GTX 1080	[10]
Chemical imaging flow cytometry		Microalgal and cancer cell detection	Label-free	CNN model	>99% for <i>Euglena gracilis</i> , >98% for white blood cells, >98% for PBMCs, >94% for Jurkat cells, and >93% for HT29 cells	NA	[17]
Reconstructing cell cycle and disease progression	Microscopy	Cell cycle of Jurkat cells/diabetic retinopathy	Labeling	CNN model	Accuracy of 98.73%	GPU	[114]

Table 2. Summary on machine learning articles on commercial IFC.

Application	Imaging	Application	Label	Algorithm	Result	Hardware	Reference
Cell imaging analysis	ImageStream or FlowSight (Amnis)	Label-free cell-cycle classification of Jurkat cells/segment image and extraction of features/multiclass machine learning	Label-free	Gradient boosting and random forest	92% accuracy	NA	[56]
Plankton classification	NA	Plankton classification	Label-free	AlexNet, GoogleNet, VGG, Resnet, DenseNet	F1 < 95%	Nvidia TitanX GPU	[106]
Microalgae classification	FlowCAM	Microalgae classification	Label-free	CNN model	88.59% accuracy	Nvidia Titan X Pascal	[105]
Plankton and coral classification	NA	Automated system for monitoring underwater ecosystems	Label-free	AlexNet, GoogleNet, VGG, Resnet, DenseNet, MobileNetV2, NasNet	F1 <= 95%	Nvidia TitanX GPU	[107]
Plankton classification	NA	Plankton image classification	Label-free	AlexNet, GoogLeNet, VGG16, ResNet, PyramidNet	86.3% accuracy	Nvidia Titan X Pascal GPU	[108]
Identification of white blood cells	ImageStream (Amnis)	Identification of white blood cells	Label-free	AdaBoost, gradient boosting, K-nearest neighbors, random forest, and support vector machine (SVM)	99% accuracy	NA	[57]
Leukemia monitoring	ImageStream (Amnis)	Label-free leukemia monitoring	Label-free	Linear SVM	98.2% accuracy	NA	[104]

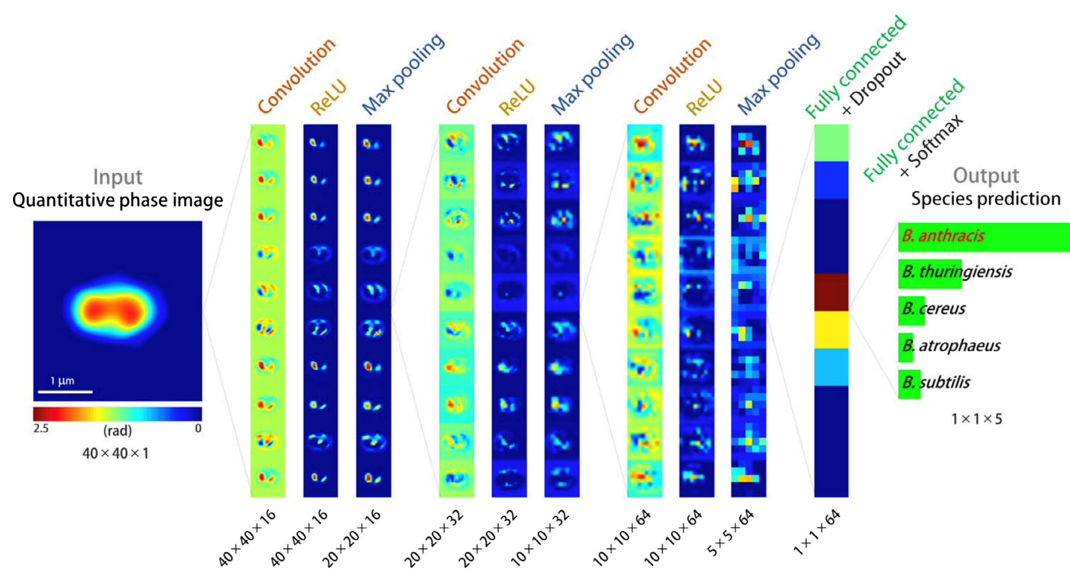


Figure 18. Holographic deep learning for anthrax detection. Reproduced with permission.^[8] Copyright 2017, American Association for the Advancement of Science.

Another example of the deep-learning-enabled portable imaging flow cytometer was developed by Gorocs et al.^[13] This device combined holographic imaging and a neural network for the rapid detection of algae in water. The resolution of the device was limited to 25 μm and was only suitable to identify large organisms.

A highly accurate algorithm based on a sophisticated dense CNN^[20] is shown in **Figure 21** for high-throughput MCF7 cancer cell detection in blood.^[21] It uses deep learning to reconstruct the cell image based on magnetically modulated lensless speckle

imaging, which uses a periodic magnetic force and lensless time-resolved holographic speckle imaging to generate target cells tagged with magnetic beads through antibodies in three dimensions. Next, the system detects those cells through a densely connected pseudo-3D CNN. It automatically detects the rare cells of interest based on the spatial-temporal features under a controlled magnetic force with a good sensing performance. But its speed is limited to 100 fps, and it has to rely on a high-performance platform with an Nvidia GeForce GTX 1080Ti GPU. High-end GPUs empower the training of complex deep neural

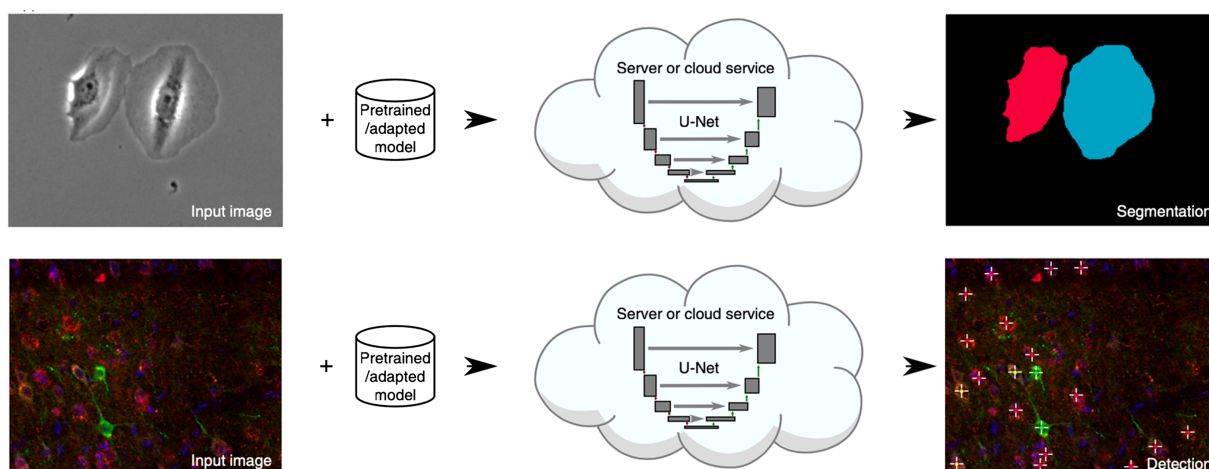


Figure 19. U-Net: deep learning for cell counting, detection, and morphometry. Top: segmentation; bottom: detection. Reproduced with permission.^[100] Copyright 2019, Springer Nature.

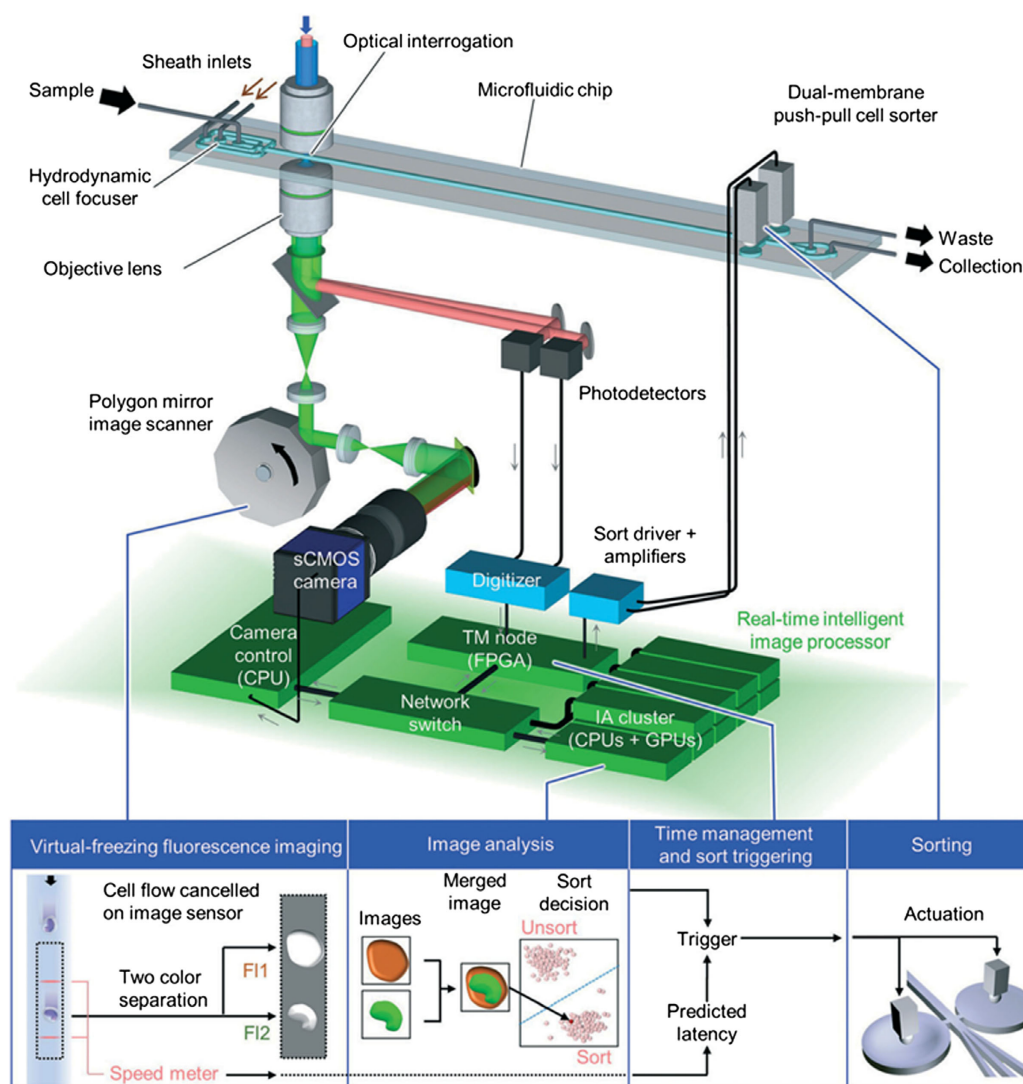


Figure 20. Intelligent image-activated cell sorting. Reproduced with permission.^[22] Copyright 2020, Royal Society of Chemistry.

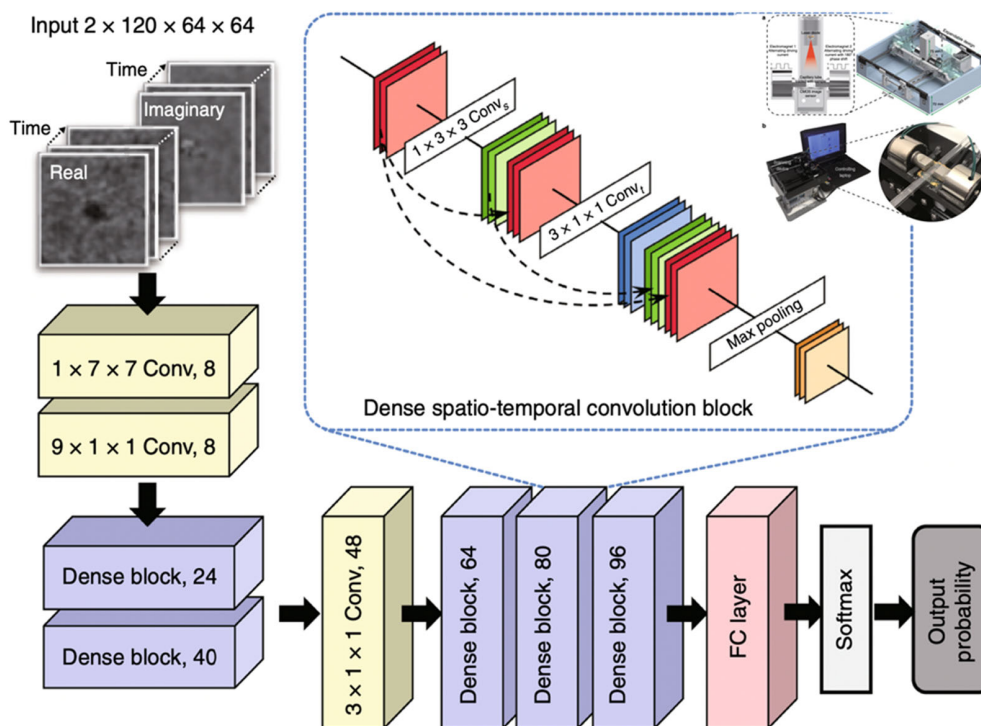


Figure 21. A classification network based on a densely connected neural network. Reproduced with permission.^[21] Copyright 2019, Springer Nature.

networks, but this is a major hurdle for mass deployment of these machine learning algorithms to commercial imaging flow cytometry for real-time object identification and classification because of the high cost and high power consumption.

Machine learning algorithms are used in commercial imaging flow cytometry for phytoplankton analysis, label-free cell-cycle identification, white blood cell identification, label-free leukemia monitoring, etc. In the early days, traditional feature extraction and classifiers,^[56,57,104] such as AdaBoost, GB, KNN, RF, and SVM, were used. With the emerging of machine learning algorithms, researchers moved their interest to deep-learning-based algorithms,^[105–108] for instance, the conventional CNN, AlexNet,^[66] VGG,^[75] GoogleNet,^[76] ResNet,^[77] DenseNet,^[20] NasNet,^[109] PyramidNet,^[110] etc. These deep-learning algorithms are optimized to achieve high prediction accuracies and require high computational resources, such as the Nvidia GTX 1080Ti, which have high power consumption, are expensive, and occupy a huge footprint when designing IIFC.

5. Conclusion

In this Review, we presented recent developments in intelligent imaging flow cytometry, such as image acquisition technologies and artificial intelligence. We introduced different imaging technologies such as a multispectral imaging system, multi-field-of-view imaging, and serial time-encoded amplified microscopy. Furthermore, we depicted the basic knowledge of visual understanding and deep learning, which is essential to understand machine learning in imaging flow cytometry. We also

discussed examples using deep learning in imaging flow cytometry by summarizing the challenges and limitations encountered.

IIFC has shown broad usages in environment monitoring, clinical diagnostics, and other biosensing applications. IIFC is expected to improve the imaging quality for revealing more distinctive features in cell images while maintaining a high throughput. Various imaging modalities have been proposed to satisfy this target, such as optofluidic time-stretch microscopy. It enables submicrometer resolution in visualizing cell structures without compromising the throughput of cell imaging.^[111] To apply deep learning for IIFC, big datasets are required to train the deep-learning model to obtain a high-accuracy classifier. Unfortunately, the development of a large-scale dataset for imaging flow cytometry is quite challenging. Labeled datasets of bio-particles requires intensive input from experts to compare with biolabeling or morphologic signals for improving the productivity and quality of the dataset labeling. Furthermore, to increase the intelligence and precision, more advanced deep learning models in general object detection need to be explored in the IIFC field. Recently, efficient deep-learning models such as MobileNetV2^[78] and SENet^[79] have attracted great interest in the research community to achieve comparable classification accuracy on cost-efficient hardware, and as IIFC is mass deployed to the board area, the cost of the whole system will be considered in the future. Efficient deep-learning models, such as those for mobile processors or low-bit deep-learning models, are worthy of interest. Furthermore, deep-learning-assisted IIFC can provide a solution for the characterization and classification of cells without the need for fluorescent labeling, which would benefit its future applications.

Acknowledgements

This work was supported by the Singapore National Research Foundation under the Competitive Research Program (NRF-CRP13-2014-01) and the Singapore Ministry of Education (MOE) Tier 3 grant (MOE2017-T3-1-001).

Conflict of Interest

The authors declare no conflict of interest.

Keywords

cell detection, deep learning, imaging flow cytometry, neural networks, optical sensing

Received: April 19, 2021

Revised: June 8, 2021

Published online: July 26, 2021

- [1] N. S. Barteneva, I. A. Vorobjev, *Imaging Flow Cytometry*, Springer, Berlin **2016**.
- [2] D. A. Basiji, W. E. Ortyu, L. Liang, V. Venkatachalam, P. Morrissey, *Clin. Lab. Med.* **2007**, 27, 653.
- [3] A. Adan, G. Alizada, Y. Kiraz, Y. Baran, A. Nalbant, *Crit. Rev. Biotechnol.* **2017**, 37, 163.
- [4] W. A. Bonner, H. R. Hulet, R. G. Sweet, L. A. Herzenberg, *Rev. Sci. Instrum.* **1972**, 43, 404.
- [5] Y. Shi, Z. Li, P. Y. Liu, B. T. T. Nguyen, W. Wu, Q. Zhao, L. K. Chin, M. Wei, P. H. Yap, X. Zhou, H. Zhao, D. Yu, D. P. Tsai, A. Q. Liu, *Adv. Photonics Res.* **2021**, 2, 2000150.
- [6] Y. Han, Y. Gu, A. C. Zhang, Y.-H. Lo, *Lab Chip* **2016**, 16:4639.
- [7] T. Blasi, H. Hennig, H. D. Summers, F. J. Theis, J. Cerveira, J. O. Patterson, D. Davies, A. Filby, A. E. Carpenter, P. Rees, *Nat. Commun.* **2016**, 7, 10256.
- [8] Y. Jo, S. Park, J. Jung, J. Yoon, H. Joo, M.-H. Kim, S.-J. Kang, M. C. Choi, S. Y. Lee, Y. Park, *Sci. Adv.* **2017**, 3, 1700606.
- [9] H. Kobayashi, C. Lei, Y. Wu, A. Mao, Y. Jiang, B. Guo, Y. Ozeki, K. Goda, *Sci. Rep.* **2017**, 7, 12454.
- [10] Y. J. Heo, D. Lee, J. Kang, K. Lee, W. K. Chung, *Sci. Rep.* **2017**, 7, 11651.
- [11] Y. Wu, A. Calis, Y. Luo, C. Chen, M. Lutton, Y. Rivenson, X. Lin, H. C. Koydemir, Y. Zhang, H. Wang, Z. Göröcs, A. Ozcan, *ACS Photonics* **2018**, 5, 4617.
- [12] N. Nitta, T. Sugimura, A. Isozaki, H. Mikami, K. Hiraki, S. Sakuma, T. Iino, F. Arai, T. Endo, Y. Fujiwaki, H. Fukuzawa, M. Hase, T. Hayakawa, K. Hiramatsu, Y. Hoshino, M. Inaba, T. Ito, H. Karakawa, Y. Kasai, K. Koizumi, S. Lee, C. Lei, M. Li, T. Maeno, S. Matsusaka, D. Murakami, A. Nakagawa, Y. Oguchi, M. Oikawa, T. Ota, et al., *Cell* **2018**, 175, 266.
- [13] Z. Gorocs, M. Tamamitsu, V. Bianco, P. Wolf, S. Roy, K. Shindo, K. Yanny, *Light Sci. Appl.* **2018**, 7, 66.
- [14] D. E. McCoy, T. Feo, T. A. Harvey, R. O. Prum, *Nat. Commun.* **2018**, 9, 1.
- [15] Y. Li, A. Mahjoubfar, C. L. Chen, K. R. Niazi, L. Pei, B. Jalali, *Sci. Rep.* **2019**, 9, 1.
- [16] P. Peuß, A. C. Box, A. Accorsi, C. Wood, A. S. Alvarado, N. Rohner, *bioRxiv* **2019**, 603035.
- [17] Y. Suzuki, K. Kobayashi, Y. Wakasaka, D. Deng, S. Tanaka, C.-J. Huang, C. Lei, C.-W. Sun, H. Liu, Y. Fujiwaki, S. Lee, A. Isozaki, Y. Kasai, T. Hayakawa, S. Sakuma, F. Arai, K. Koizumi, H. Tezuka, M. Inaba, K. Hiraki, T. Ito, M. Hase, S. Matsusaka, K. Shiba, K. Suga, M. Nishikawa, M. Jona, Y. Yatomi, Y. Yalikun, Y. Tanaka, et al., *Proc. Natl. Acad. Sci.* **2019**, 116, 15842.
- [18] A. G. Memon, Y. Xing, X. Zhou, R. Wang, L. Liu, S. Zeng, M. He, M. Ma, J. *Hazard Mater.* **2020**, 384, 120948.
- [19] A. Lumini, L. Nanni, G. Maguolo, *Appl. Comput. Inf.* **2019**, 51, 33.
- [20] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**.
- [21] J. Lim, W. Liang, A. A. Savchenkov, A. B. Matsko, L. Maleki, C. W. Wong, *Light Sci. Appl.* **2019**, 8, 1.
- [22] A. Isozaki, H. Mikami, H. Tezuka, H. Matsumura, K. Huang, M. Akamine, K. Hiramatsu, T. Iino, T. Ito, H. Karakawa, Y. Kasai, Y. Li, Y. Nakagawa, S. Ohnuki, T. Ota, Y. Qian, S. Sakuma, T. Sekiya, Y. Shirasaki, N. Suzuki, E. Tayyabi, T. Wakamiya, M. Xu, M. Yamagishi, H. Yan, Q. Yu, S. Yan, D. Yuan, W. Zhang, Y. Zhao et al., *Lab Chip* **2020**, 20, 2263.
- [23] C. E. Pedreira, E. S. Costa, Q. Lecrevisse, J. J. van Dongen, A. Orfao, *Trends Biotechnol.* **2013**, 31, 415.
- [24] Y. Wang, F. Hammes, K. De Roy, W. Verstraete, N. Boon, *Trends Biotechnol.* **2010**, 28, 416.
- [25] R. M. Zucker, E. J. Massaro, K. M. Sanders, L. L. Degn, W. K. Boyes, *Cytometry, Part A* **2010**, 77, 677.
- [26] H. M. Shapiro, *Practical Flow Cytometry*, John Wiley & Sons, Hoboken, NJ **2005**.
- [27] K. M. McKinnon, *Curr. Protoc. Immunol.* **2018**, 120, 5.1. 1.
- [28] M. G. Macey, *Flow Cytometry*, Springer, Berlin **2007**.
- [29] E. Schonbrun, S. S. Gorthi, D. Schaak, *Lab Chip* **2012**, 12, 268.
- [30] E. D. Diebold, B. W. Buckley, D. R. Gossett, B. Jalali, *Nat. Photonics* **2013**, 7, 806.
- [31] H. Mikami, H. Kobayashi, S. Hamad, Y. Ozeki, K. Goda, in *CLEO: Applications and Technology: Optical Society of America*, **2016**.
- [32] R. Hain, C. J. Kähler, C. Tropea, *Exp. Fluids* **2007**, 42, 403.
- [33] M. McClish, R. Farrell, R. Myers, F. Olschner, G. Entine, K. S. Shah, *Nucl. Instrum. Methods Phys. Res. Sect. A* **2006**, 567, 36.
- [34] M. B. Clark, T. R. Mercer, G. Bussotti, T. Leonardi, K. R. Haynes, J. Crawford, M. E. Brunck, K. L. Cao, G. P. Thomas, W. Y. Chen, R. J. Taft, L. K. Nielsen, A. J. Enright, J. S. Mattick, M. E. Dinger, *Nat Methods* **2015**, 12, 1.
- [35] D. A. Basiji, W. E. Ortyu, L. Liang, V. Venkatachalam, P. Morrissey, *Clin. Lab. Med.* **2007**, 27, 653.
- [36] G. Elliott, *Comb. Chem. High Throughput Screening* **2009**, 12, 849.
- [37] D. A. Basiji, in *Imaging Flow Cytometry* (Eds: N. S. Barteneva, I. A. Vorobjev), Springer, Berlin **2016**, pp. 13–21.
- [38] S. S. Gorthi, D. Schaak, E. Schonbrun, *Opt. Express* **2013**, 21, 5164.
- [39] Y. Han, Y.-H. Lo, *Sci. Rep.* **2015**, 5, 13267.
- [40] Y. Han, Y.-H. Lo, in *High-Speed Biomedical Imaging and Spectroscopy: Toward Big Data Instrumentation and Management*, International Society for Optics and Photonics, Bellingham, WA **2016**.
- [41] C. K. Sieracki, M. E. Sieracki, C. S. Yentsch, *Mar. Ecol. Prog. Ser.* **1998**, 168, 285.
- [42] N. J. Poulton, in *Imaging Flow Cytometry* Springer, Berlin **2016**, pp. 237–247.
- [43] R. J. Olson, H. M. Sosik, *Limnol. Oceanogr. Methods* **2007**, 5, 195.
- [44] B. G. Batchelor, *Machine Vision Handbook*, Springer, Berlin **2012**.
- [45] T. Taulli, M. Oni, *Artificial Intelligence Basics*, Springer, Berlin **2019**.
- [46] M. Nixon, A. Aguado, *Feature Extraction and Image Processing for Computer Vision*, Academic press, Cambridge, MA **2019**.
- [47] P. Viola, M. Jones, *CVPR* **2001**, 1, 3.
- [48] P. Felzenszwalb, D. McAllester, D. Ramanan, in *2008 IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2008**.
- [49] N. Dalal, B. Triggs, in *IEEE Computer Society*, IEEE, Piscataway, NJ **2005**.

- [50] A. Satpathy, X. Jiang, H.-L. Eng, *IEEE Trans. Image Process.* **2014**, 23, 287.
- [51] D. G. Lowe, *Int. J. Comput. Vision* **2004**, 60, 91.
- [52] J. Ren, X. Jiang, J. Yuan, *IEEE Trans. Image Process.* **2013**, 22, 4049.
- [53] A. Satpathy, A. Xudong Jiang, A. How-Lung Eng, *IEEE Trans. Image Process.* **2014**, 23, 1953.
- [54] A. Natekin, A. Knoll, *Front. Neurobiol.* **2013**, 7, 21.
- [55] G. Biau, E. Scornet, *Test* **2016**, 25, 197.
- [56] H. Hennig, P. Rees, T. Blasi, L. Kamentsky, J. Hung, D. Dao, A. E. Carpenter, A. Filby, *Methods* **2017**, 112, 201.
- [57] M. Nassar, M. Doan, A. Filby, O. Wolkenhauer, D. K. Fogg, J. Piasecka, C. A. Thornton, A. E. Carpenter, H. D. Summers, P. Rees, H. Hennig, *Cytometry, Part A* **2019**, 95, 836.
- [58] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, *Neural Comput.* **2001**, 13, 1443.
- [59] X. Jiang, *IEEE Signal Process. Mag.* **2011**, 28, 16.
- [60] R. De Maesschalck, D. Jouan-Rimbaud, D. L. Massart, *Chemom. Intell. Lab. Syst.* **2000**, 50, 1.
- [61] M. A. Peterson, B. S. Gibson, *Cognit. Psychol.* **1993**, 25, 383.
- [62] M. A. Turk, A. P. Pentland, in *Proc. 1991 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society, Maui, HI, USA **1991**.
- [63] E. C. Greanias, P. F. Meagher, R. J. Norman, P. Essinger, *IBM J. Res. Dev.* **1963**, 7, 14.
- [64] J. M. Ferryman, A. D. Worrall, G. D. Sullivan, K. D. Baker, in *BMVC*, Citeseer **1995**.
- [65] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, *Proc. IEEE* **1998**, 86, 2278.
- [66] A. Krizhevsky, I. Sutskever, G. E. Hinton, **2012**.
- [67] Z.-Q. Zhao, P. Zheng, S.-T. Xu, X. Wu, *IEEE Trans. Neural Networks Learn. Syst.* **2019**, 30, 3212.
- [68] K. Noda, Y. Yamaguchi, K. Nakada, H. G. Okuno, T. Ogata, *Appl. Intell.* **2015**, 42, 722.
- [69] M. Wainberg, D. Merico, A. Delong, B. J. Frey, *Nat. Biotechnol.* **2018**, 36, 829.
- [70] T. Young, D. Hazarika, S. Poria, E. Cambria, *IEEE Comput. Intell. Mag.* **2018**, 13, 55.
- [71] S. Ruder, arXiv preprint arXiv:1609.04747 **2016**.
- [72] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press Cambridge, MA, **2016**.
- [73] K. Fukushima, S. Miyake, in *Competition and Cooperation in Neural Nets*, Springer, Berlin **1982**, pp. 267–285.
- [74] D. C. Cireşan, U. Meier, L. M. Gambardella, J. Schmidhuber, *Neural Comput.* **2010**, 22, 3207.
- [75] K. Simonyan, A. Zisserman, arXiv preprint arXiv:1409.1556 **2014**.
- [76] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, **2015**.
- [77] K. He, X. Zhang, S. Ren, J. Sun, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2016**.
- [78] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2018**, 1, 4510.
- [79] J. Hu, L. Shen, G. Sun, **2018**.
- [80] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, Big Transfer (BiT): General Visual Representation Learning Supplementary Material, www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123500477-sup.pdf.
- [81] R. K. Srivastava, K. Greff, J. Schmidhuber, arXiv preprint arXiv:1505.00387 **2015**.
- [82] F. Chollet, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**, pp. 1800–1807.
- [83] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, arXiv preprint arXiv:1704.04861 **2017**.
- [84] S. Luo, Y. Shi, L. K. Chin, Y. Zhang, B. Wen, Y. Sun, B. T. T. Nguyen, G. Chierchia, H. Talbot, T. Bourouina, X. Jiang, A.-Q. Liu, *RSC Adv.* **2021**, 11, 17603.
- [85] J. Lu, J. Hu, J. Zhou, in *IEEE Signal Process. Mag.* **2017**, 34, 76.
- [86] E. Hoffer, N. Ailon, in *Int. Workshop on Similarity-Based Pattern Recognition*, Springer, Berlin **2015**.
- [87] S. Luo, K. T. Nguyen, B. T. T. Nguyen, S. Feng, Y. Shi, A. Elsayed, Y. Zhang, X. Zhou, B. Wen, G. Chierchia, et al., *Cytometry, Part A*, n/a.
- [88] S. Luo, Y. Zhang, K. T. Nguyen, S. Feng, Y. Shi, Y. Liu, P. Hutchinson, G. Chierchia, H. Talbot, T. Bourouina, X. Jiang, A. Q. Liu, et al., *Micromachines* **2020**, 11, 1084.
- [89] D. C. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, *Med. Image Comput. Comput. Assist. Interv.* **2013**, 16, 411.
- [90] J. Hung, A. Carpenter, **2017**.
- [91] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, 40, 834.
- [92] K. He, G. Gkioxari, P. Dollar, R. Girshick, *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, 42, 386.
- [93] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, Z. He, in *2017 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, IEEE, Piscataway, NJ **2017**.
- [94] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P. H. Torr, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2019**.
- [95] N. Wojke, A. Bewley, D. Paulus, in: *2017 IEEE Int. Conf. on Image Processing (ICIP)*, IEEE, Piscataway, NJ **2017**.
- [96] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, B. Leibe, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2019**.
- [97] Y. Li, L. Shen, S. Yu, *IEEE Trans. Med. Imaging* **2017**, 36, 1561.
- [98] E. Shelhamer, J. Long, T. Darrell, *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, 39, 640.
- [99] O. Ronneberger, P. Fischer, T. Brox, In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015*. (Eds. N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi), Springer, Cham. **2015**, vol 9351, pp. 234–241.
- [100] T. Falk, D. Mai, R. Besch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, O. Ronneberger, *Nat. Methods* **2019**, 16, 67.
- [101] Y. Shi, T. Zhu, T. Zhang, A. Mazzulla, D. P. Tsai, W. Ding, A. Q. Liu, G. Cipparrone, J. J. Sáenz, C.-W. Qiu, *Light Sci. Appl.* **2020**, 9, 62.
- [102] Y. Shi, S. Xiong, L. K. Chin, J. Zhang, W. Ser, J. Wu, T. Chen, Z. Yang, Y. Hao, B. Liedberg, P. H. Yap, D. P. Tsai, C.-W. Qiu, A. Q. Liu, *Sci. Adv.* **2018**, 4, eaao0773.
- [103] H. Mikami, M. Kawaguchi, C.-J. Huang, H. Matsumura, T. Sugimura, K. Huang, C. Lei, S. Ueno, T. Miura, T. Ito, K. Nagasawa, T. Maeno, H. Watarai, M. Yamagishi, S. Uemura, S. Ohnuki, Y. Ohya, H. Kurokawa, S. Matsusaka, C.-W. Sun, Y. Ozeki, K. Goda, *Nat. Commun.* **2020**, 11, 1.
- [104] M. Doan, M. Case, D. Masic, H. Hennig, C. McQuin, J. Caicedo, S. Singh, A. Goodman, O. Wolkenhauer, H. D. Summers, D. Jamieson, F. W. Delft, A. Filby, A. E. Carpenter, P. Rees, J. Irving, *Cytometry, Part A* **2020**, 97, 407.
- [105] I. Correa, P. Drews, S. Botelho, M. S. de Souza, in *2017 16th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*: IEEE, Piscataway, NJ **2017**.
- [106] A. Lumini, L. Nanni, *Ecol. Inf.* **2019**, 51, 33.
- [107] A. Lumini, L. Nanni, G. Maguolo, *Appl. Comput. Inf.* **2020**, <https://doi.org/10.1016/j.aci.2019.11.004>.

- [108] J. Liu, A. Du, C. Wang, Z. Yu, H. Zheng, B. Zheng, H. Zhang, in *2018 Oceans–MTS/IEEE Kobe Techno–Oceans (OTO)*, IEEE, Piscataway, NJ **2018**.
- [109] X. Qin, Z. Wang, arXiv preprint arXiv:1912.03151 **2019**.
- [110] D. Han, J. Kim, J. Kim, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**.
- [111] C. Lei, H. Kobayashi, Y. Wu, M. Li, A. Isozaki, A. Yasumoto, H. Mikami, T. Ito, N. Nitta, T. Sugimura, M. Yamada, Y. Yatomi, D. Di Carlo, Y. Ozeki, K. Goda, *Nat. Protoc.* **2018**, 13, 1603.
- [112] N. Meng, E. Y. Lam, K. K. Tsia, H. K. So, *IEEE J. Biomed. Health Inf.* **2018**, 23, 2091.
- [113] L. Wang, N. J. Long, L. Li, Y. Lu, M. Li, J. Cao, Y. Zhang, Q. Zhang, S. Xu, Z. Yang, C. Mao, M. Peng, *Light Sci. Appl.* **2018**, 7, 1.
- [114] P. Eulenberg, N. Köhler, T. Blasi, A. Filby, A. E. Carpenter, P. Rees, F. J. Theis, F. A. Wolf, *Nat. Commun.* **2017**, 8, 1.



Shaobo Luo received his Ph.D. degree in electronics, optronics, and systems from University Gustave Eiffel, France, in 2021. Currently, he is a director of system R&D of Shanghai Gene Sense Biotech Co., Ltd., and a senior visiting scholar at School of Microelectronics, Southern University of Science and Technology. He has innovated many intelligent algorithms for high-throughput bioparticle recognition and sizing, and also developed revolutionary intelligent systems. His research interests include artificial intelligence, embedded systems, high-performance computing, and robotics.



Yuzhi Shi received his Ph.D. degree in mechanical engineering from Xi'an Jiaotong University in 2018. He joined Nanyang Technological University in 2013 and became a research fellow in the School of Electrical and Electronic Engineering in 2018. He has innovated many revolutionary optofluidic manipulation techniques, such as bacteria and virus screening and binding, chiral particle sorting, multipole-assisted giant optical forces on Mie-resonant particles, etc. His research interests include the optical trapping and sorting of nanoparticles and biomolecules, optofluidic devices, extraordinary light-matter interactions, and dielectric nanophotonics.



Lip Ket Chin received his Ph.D. degree in electrical and electronic engineering from Nanyang Technology University (NTU) in Singapore. He is currently a senior research fellow at NTU and a research fellow at the Harvard Medical School and Massachusetts General Hospital, USA. He has innovated many optofluidic lab-on-a-chip devices, and also developed revolutionary biosensors, such as a portable cell analyzer for cancer diagnosis, a nanoplasmonic molecular biosensor, and virus manipulation toolkits. His current research interest focuses on the development of nanophotonic and nanoplasmonic platforms for biomedical sensing and imaging, cancer cell studies, virus studies, etc.



Xudong Jiang is currently a professor at the School of Electrical and Electronic Engineering, Nanyang Technological University. He received his Ph. D. degree from Helmut Schmidt University, Hamburg, Germany, and B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China (UESTC). He holds seven patents and has authored over 150 articles with over 40 articles in the IEEE journals. His current research interests include image processing, pattern recognition, computer vision, machine learning, and biometrics.



Tarik Bourouina (Senior Member, IEEE) received his M.Sc. degree in physics, M.Eng. degree in optoelectronics, and Ph.D. and H.D.R. degrees from Université Paris-Saclay, Gif-sur-Yvette, France, in 1987, 1988, 1991, and 2000, respectively. He is currently a full professor in physics with the ESIEE Paris School of Engineering, a founding member of the Université Gustave Eiffel, Marne-la-Vallée, France, where he has acted as Associate Vice-President for Research since 2020. He has authored or coauthored more than 350 scientific publications in international journals and conferences and has 13 patents. His research activity is conducted within the ESYCOM Lab (CNRS UMR 9007), where his current interests include photonic MEMS, micro-optofluidics, and metasurfaces.



Ai Qun Liu is currently a professor at the School of Electrical and Electronic Engineering, Nanyang Technological University. He received his Ph.D. degree from National University of Singapore (NUS) in 1994. He serves as an editor and editorial board member of several journals. He is the author of over 150 publications, including peer-reviewed journal articles and two books. Also, he is an OSA Fellow, RCS Fellow, and SPIE Fellow. His research interest includes optofluidics, silicon photonics, and quantum devices.