



HAL
open science

Machine Learning on High-Throughput Bioparticles Image Recognition and Sizing

Shaobo Luo

► **To cite this version:**

Shaobo Luo. Machine Learning on High-Throughput Bioparticles Image Recognition and Sizing. Electronics. Université Gustave Eiffel, 2021. English. NNT : 2021UEFL2013 . tel-03738133

HAL Id: tel-03738133

<https://theses.hal.science/tel-03738133>

Submitted on 25 Jul 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.



École Doctorale MSTIC

**Mathématiques et Sciences et Technologies de l'Information et de la
Communication**

Thèse

Présentée pour l'obtention du grade de DOCTEUR

DE L'UNIVERSITE GUSTAVE EIFFEL
par

Shaobo LUO

Apprentissage automatique sur la reconnaissance et le dimensionnement d'images de bioparticules à haut débit

Spécialité : Électronique, Optronique & Systèmes

Jury

Hugues TALBOT, Professeur, Université Paris-Saclay, France (Président)
Etienne DECENCIERE, Directeur de Recherches, PSL Université, France (Rapporteur)
Diaa KHALIL, Professeur, Ain-Shams University, Egypt (Rapporteur)
Elodie PUYBAREAU, Professeur Assistant, EPITA, France (Examinateur)
Xudong JIANG, Professeur, Nanyang Technological University, Singapore (Co-encadrant)
Giovanni CHIERCHIA, Professeur Assistant, Université Gustave Eiffel, France (Co-encadrant)
Tarik BOUROUINA, Professeur, Université Gustave Eiffel, France (Directeur de thèse)
Ai-Qun LIU, Professeur, Nanyang Technological University, Singapore (Co-Directeur de thèse)



LABORATOIRE ÉLECTRONIQUE, SYSTÈMES
DE COMMUNICATION ET MICROSYSTÈMES

Sous la co-tutelle de :

CNAM
CNRS
UNIVERSITÉ GUSTAVE EIFFEL

Thèse effectuée au sein du **Laboratoire ESYCOM (UMR 9007 CNRS)**

Électronique, SYstèmes de Communication

& Microsystèmes

de l'Université Gustave Eiffel
5 bd Descartes, Cité Descartes,

Champs-sur-Marne 77454 Marne-la-Vallée cedex 2

France

Machine Learning on High-Throughput Bioparticles Image Recognition and Sizing

Acknowledgements

First and foremost, I would like to express my sincere gratitude and deep appreciations to my supervisors: Professor Tarik Bourouina, Professor Giovanni Chierchia, Professor Hugues Talbot, Professor Xudong Jiang, and Professor Ai-Qun Liu who gave me the great opportunity for this PhD project in University of Gustave Eiffel. During the whole project, they also gave me the unwavering guidance and constant support. They encouraged and innovated me with the methodologists and ideas when I did my research. I also express my gratitude to Dr Yuzhi Shi, Professor Yi Zhang, and Professor Bihan Wen for giving me the valuable guidance, suggestion and support. I also would like to thanks our group members: Dr Gong Zhang, Dr Nguyen Kim Truc, Dr Nguyen Thi Thanh, Dr Hailong Li, Dr Jingbo Zhang, Dr Yu Jiaqing, Dr Yang Liu, Mr Backiam Natesan Prakash, Miss Lipika Mahato, Mr Bin Li, Dr Shilun Feng, Dr Junfeng Wu, Dr Ping Su, Dr Lim Change Nong, Dr Jun Zou from Quantum Science & Engineering Centre (QSec), Nanyang Technological University of Singapore, Mr Ahmed Elsayed, Mazen Erfan from University of Gustave Eiffel and Dr Yixin Wang, Dr Hui Dong, Dr Jun Li, Dr Ying Sun, Dr Kong Wah Wan, Dr Juanjuan Hu, Mr Beng Sing Tan, Mr Raymond Keh from Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR) for their kind help and support during my research work. Furthermore, I am grateful to University of Gustave Eiffel, University of Paris-Est (Paris-Est Sup), and Nanyang Technological University of Singapore for supporting my graduate studies for the entire PhD journey. Last but not least, I would like to thank my family for supporting me with space and time to finish my studies.

Luo Shaobo

Summary

Imaging flow cytometry is a high-throughput tool widely used for bioparticle analysis in various applications. However, the vast number of images generated by imaging flow cytometry imposes a great challenge for data analysis. Although various learning algorithms were optimized to achieve high prediction accuracies, they overlooked the trade-off between speed and hardware requirements. This causes a major hurdle for mass deployment of these learning algorithms to commercial devices for high-throughput bioparticle analysis due to its high cost and high-power consumption. Moreover, rare bioparticle detection is still a significant challenge because the representative training images are hard to collect and the input images in inference may be different from the training images. Furthermore, the size measurement algorithms do not always lead to accurate sizing without size level calibration. In this thesis, we have developed an efficient neural network named MCellNet for a rapid, accurate and high-throughput detection of label-free *Cryptosporidium*, *Giardia* and other pollutants captured by imaging flow cytometry. MCellNet achieved a classification accuracy over 99.6% and a processing speed of 346+ images per second on an embedded platforms, outperforming MobileNetV2 (251 frames per second) with a similar classification accuracy. In addition, deep metric learning for rare bioparticle detection was also studied. Deep metric learning based classification algorithm enhanced the accuracy with traditional deep-learning-based features as it encodes more semantic information into the network, thus deep metric learning performs better than simple classification of deep features and it is well adapted for inlier bioparticle detection. Furthermore, a machine-learning-based pipeline was established for the high accuracy bioparticle sizing. The pipeline

consisted of an image segmentation module for measuring the pixel size of the bioparticle and a machine learning model for accurate pixel-to-size conversion. The sizing algorithm showed significantly more accuracy and promised greater potential in a wide range of bioparticle sizing applications. The proposed methods could also be potentially applied to other high-throughput and real-time bioparticle analysis for biomedical diagnosis, environmental monitoring, and other bioparticle detection applications.

Keywords:

Deep learning, Bioparticle classification, Convolution neural network, Deep metric learning, Bioparticle sizing, Bioparticle images

Résumé

L'imagerie par cytométrie de flux est un outil à très haute cadence largement utilisé pour l'analyse des bioparticules. Néanmoins, le grand nombre d'images générées pose un grand défi pour l'analyse rapide des données. Bien que divers algorithmes d'apprentissage aient été optimisés pour atteindre une précision de prédiction élevée, ils ont négligé le compromis entre la vitesse et les exigences matérielles, ce qui a causé un obstacle majeur au déploiement de masse de ces algorithmes d'apprentissage sur des appareils commerciaux pour l'analyse des bioparticules à haut débit, ceci en raison de son coût et sa consommation d'énergie élevée. Dans cette thèse, nous avons développé un réseau neuronal efficace, appelé MCellNet, pour une approche rapide, précise et à haut débit pour la détection de bioparticules sans marquage fluorescent, adapté à la cytométrie de flux. MCellNet a atteint une précision de classification de plus de 99,6% et une vitesse de traitement de plus de 346 images par seconde dans les plates-formes intégrées, surpassant MobileNetV2 (251 images par seconde) avec une précision de classification similaire. En outre, l'apprentissage métrique profond pour la détection de bioparticules rares a également été étudié. En outre, un pipeline basé sur l'apprentissage automatique est établi pour la classification en taille et à haute précision des bioparticules. Le pipeline se compose d'un module de segmentation d'image pour mesurer la taille de pixel de la bioparticule et d'un modèle d'apprentissage automatique pour une conversion précise pixel-taille. L'algorithme a montré une capacité de classification en taille nettement plus précise et présente un grand potentiel pour une large gamme d'applications pour la détermination des tailles de bioparticules. Les méthodes proposées pourraient également être potentiellement appliquées à d'autres analyses de bioparticules à haut débit et en temps réel pour le diagnostic biomédical et la surveillance environnementale.

Mots clés:

Apprentissage profond, classification des bioparticules, réseau de neurones à convolution, apprentissage métrique profond, dimensionnement des bioparticules, images de bioparticules

Table of Contents

Acknowledgements	i
Summary	ii
List of Figures	x
List of Tables	xviii
Nomenclature	xx
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Objectives	5
1.3 Major Contributions	7
1.4 Organization of the Thesis	9
Chapter 2 Literature Survey	11
2.1 Introduction	12
2.2 Machine Learning	14
2.2.1 Machine Vision and Image Analysis	14
2.2.2 Traditional Machine Learning	14
2.2.3 Deep Learning	21
2.3 Imaging Flow Cytometry	35
2.4 Deep Learning Applications	40
2.5 Summary	52
Chapter 3 Deep Learning-Enabled High-Speed Bioparticle Detection	55
3.1 Introduction	56
3.2 Model for High-speed Bioparticle Detection	63
3.2.1 Overall Structure of MCellNet	63

3.2.2	Training of the Model	66
3.2.3	Optimization of the Model.....	67
3.3	Experimental Results and Discussions.....	69
3.3.1	Image Acquisition of Bioparticles using Imaging Flow Cytometry..	69
3.3.2	Evaluation Metrics	72
3.3.3	Multiclass Classification.....	82
3.3.4	Binary Classification.....	88
3.4	Summary	92
Chapter 4	Deep Metric Learning-Enabled Rare Bioparticle Detection.....	93
4.1	Introduction	94
4.2	Model for Rare Bioparticle Detection.....	98
4.2.1	The Fundamental of Deep Metric Learning.....	98
4.2.2	Deep Metric Learning for Rare Bioparticle Detection	99
4.2.3	Training of Deep Metric Learning.....	102
4.2.4	Deep Metric Learning based Model	104
4.3	Experimental Results and Discussions.....	107
4.3.1	Profiling of Bioparticles.....	107
4.3.2	Bioparticle Classification Evaluation	109
4.3.3	Model Verification using Spiked Samples	113
4.4	Summary	116
Chapter 5	Machine Learning-Based Pipeline for Bioparticle Sizing	117
5.1	Introduction	118

5.2	The Model of High Accuracy Bioparticle Sizing.....	125
5.2.1	Segmentation and Pixel Measurement.....	125
5.2.2	Conversion of Pixel Size to Physical Size	129
5.3	Experimental Results and Discussions.....	133
5.3.1	Performance Evaluation.....	133
5.3.2	Segmentation and Pixel Size Measurement.....	135
5.3.3	Physical Size Measurement	135
5.4	Summary	142
Chapter 6	Concluding Remarks.....	143
6.1	Conclusions	143
6.2	Recommendations	146
6.2.1	Efficient Neural Networks Design.....	146
6.2.2	Network Improving.....	146
	Bibliography	149
	Author's Publications.....	175

List of Figures

Figure 1-1: Categories of <i>Cryptosporidium</i> , <i>Giardia</i> and other viruses and bacteria.	2
Figure 2-1: Overview of intelligent imaging flow cytometry.....	12
Figure 2-2: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlaid on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose. Reproduced with permission from IEEE [6].....	15
Figure 2-3: Detections obtained with a single component person model. The model is defined by a coarse root filter (a), several higher resolution part filters (b) and a spatial model for the location of each part relative to the root (c). The filters specify weights for histogram of oriented gradients features. Their visualization show the positive weights at different orientations. The visualization of the spatial models reflects the “cost” of placing the center of a part at different locations relative to the root. Reproduced with permission from IEEE [7].....	16
Figure 2-4: Working principle of histograms of oriented Gradients.	17
Figure 2-5: Local Binary Patterns. (a) The procedure diagram on local binary patterns histograms. (b) The diagram on how to calculate Local Binary Patterns.....	18
Figure 2-6: The architecture of LeNet-5 neural network. A convolutional neural network for digital recognition. Reproduced with permission from IEEE [2].	21
Figure 2-7: Comparison between (a) traditional handcrafted approach and (b) deep learning.	22
Figure 2-8: Convolutional neural network.....	23

Figure 2-9: Layers of convolutional neural network. (a) Convolutional operation. (b) Rectified Linear Unit (ReLU). (c) Max-pooling operation.	24
Figure 2-10: ImageNet challenge.	26
Figure 2-11: Residual learning building block.	28
Figure 2-12: Examples of network architecture for ImageNet. the VGG-19 model is in the left, a plain network with 34 parameter layers is in the middle, and a residual network with 34 parameter layers in the right. The dotted shortcuts increase dimensions. Reproduced with permission from IEEE [81].	32
Figure 2-13: (a) Depthwise convolution. (b) Pointwise convolution with 256 kernels.	33
Figure 2-14: General pipeline of deep metric learning. It includes embedding network for extracting features to latent embedding space, sampling strategies, and loss functions.	34
Figure 2-15: 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 from Royal Society of Chemistry [33].	36
Figure 2-16: ImageStreamX Mk II.	38
Figure 2-17: FlowCam System.	39
Figure 2-18: Submersible Imaging FlowCytobot.	40
Figure 2-19: R-MOD (Real-time Moving Object Detector) system. Reproduced with permission from Nature Research [3].	42
Figure 2-20: Classification and Segmentation. (a) Classification, (b) Segmentation.	43
Figure 2-21: U-Net. Reproduced with permission from Nature Research [1].	44

Figure 2-22: Holographic deep learning. Reproduced with permission from American Association for the Advancement of Science [8].	45
Figure 2-23: U-Net: deep learning for cell counting, detection, and morphometry. Reproduced from ref. [1] with permission from Nature Research.	46
Figure 2-24: Intelligent Image-Activated Cell Sorting. Reproduced with permission from Royal Society of Chemistry [4].	47
Figure 2-25: A classification network based on densely connected neural network. Reproduced from ref. [5] with permission from Nature Research.	48
Figure 3-1: Collected bioparticles which plotted by length and florescent intensity.	57
Figure 3-2: The selected CG area (<i>Cryptosporidium</i> and <i>Giardia</i>) with intensity by expert.	58
Figure 3-3: The fine selected C (<i>Cryptosporidium</i>) area and G (<i>Giardia</i>) area.	59
Figure 3-4: The histogram graph of the selected <i>Cryptosporidium</i> and <i>Giardia</i>	60
Figure 3-5: Overview of MCellNet, a deep neural network assists the imaging flow cytometry (Amnis® ImageStream®X Mk II) in <i>Cryptosporidium</i> and <i>Giardia</i> detection. The system consists of laser, flow cytometry, imaging system, image database and deep neural network.	60
Figure 3-6: MCellNet architecture, which contains one convolutional layer, 6 inverted residual blocks, one flattened layer, and one fully connected layer. Each IRB forms a shortcut between the bottlenecks to perform identity mapping.	63
Figure 3-7: Raw image sequence of bioparticles. Each line represents one bioparticle's images. From left to right is fluorescent image, bright-field image, and side scattering image.....	69

Figure 3-8: Bioparticle image dataset. Each row represents one type of bioparticles. From the top to bottom are *Cryptosporidium*, *Giardia*, microbeads ($12\mu m$, $4\mu m$, $5.64\mu m$, $8\mu m$, $15\mu m$, and $3\mu m$.), and natural pollutants. All sub figures share the same scale bar.71

Figure 3-9: Single bioparticle image dataset. Each block represents one type of bioparticle. From the top to bottom are (a) *Cryptosporidium*, (b) *Giardia*, microbeads with size of (c) $1.54\mu m$, (d) $3\mu m$, (e) $4\mu m$, (f) $4.6\mu m$, (g) $5\mu m$, (h) $5.64\mu m$, (i) $8\mu m$, (j) $10\mu m$, (k) $12\mu m$, and (l) $15\mu m$, and (m) natural pollutants. All figures share the same scale bar in (a).79

Figure 3-10: Data Distribution on Gradient RMS. (a) The normalized histogram of Gradient RMS. (b) The example images with of lesser RMS (blur) and high RMS (clear).80

Figure 3-11: The normalized confusion matrix on multiclass classification task. (a) MobileNetV2 and (b) Our proposed MCellNetM. 13 classes of output include naturally existing pollutants, *Cryptosporidium*, *Giardia*, and beads: $1.54\mu m$, $3\mu m$, $4\mu m$, $4.6\mu m$, $5\mu m$, $5.64\mu m$, $8\mu m$, $10\mu m$, $12\mu m$, and $15\mu m$. The color grading represents the level of agreement between the prediction and the ground truth with white (1) being full agreement and black (0) being no agreement. In an ideal case with no misclassification, all boxes along the diagonal from the top left to bottom right have a value of 1 and appear white, and all the rest boxes have a value of zero and appear black. With misclassification, the non-diagonal boxes would have non-zero values and appear red.84

Figure 3-12: The t-SNE visualization on the IRB0 to IRB5 (a)-(f) of MCellNetM in multiclass classification. Different color is the different target class of

<i>Cryptosporidium</i> , <i>Giardia</i> , microbeads (1.54 μm , 3 μm , 4 μm , 4.6 μm , 5 μm , 5.64 μm , 8 μm , 10 μm , 12 μm , and 15 μm .), and natural pollutants.....	87
Figure 3-13: The False Negative-False Positive rate curve on the binary classification task. The diagonal dashed is the EER. Natural pollutant images are class 0, and all <i>Cryptosporidium</i> and <i>Giardia</i> images are class 1.....	89
Figure 3-14: The t-SNE visualization on the IRB0 to IRB5 (a)-(f) of MCellNetB in binary classification. Natural pollutant images are class 0 (green), and all <i>Cryptosporidium</i> and <i>Giardia</i> images are class 1 (red).....	90
Figure 4-1: Novel examples in the inference time. The conventional deep neural networks often make wrong predictions and do so confidently on some novel examples when the images are not seen in the training dataset. Such as the pollutions are predicted as <i>Giardia</i> or <i>Cryptosporidium</i> with confidence level > 99.99%.	94
Figure 4-2: Conventional deep classification. In deep classification, the model only studies a boundary.	95
Figure 4-3: Deep metric learning problem. In deep metric learning, the model studies a more generative representation with similar classes are close and the unsimilar classes are far away.....	96
Figure 4-4: Embedding network structure.	100
Figure 4-5: Residual blocks in the embedding network.	101
Figure 4-6: The structure of Siamese network for training deep metric learning. The twin networks share same network parameters. A loss function is attached to this twin network to regularize the network.	102
Figure 4-7: Deep metric learning based classification. The unknown bioparticle is classified to correspond classes. Class label is assigned to classify the unknown particles by the closed cluster center (red). Confidence level is used to present the	

similarity of unknown particles to certain databased collected by *Cryptosporidium* and *Giardia* samples.104

Figure 4-8: Overview of system setup. (a) A deep metric neural network that enables intelligent imaging flow cytometry (Amnis® ImageStream®X Mk II) for rare bioparticle detection. The system consists of laser, flow cytometry, imaging system, image database, and deep neural network, etc. (b) Images are collected into a database.107

Figure 4-9: Bioparticle image dataset. Each row represents one type of bioparticle. From the top to bottom are *Cryptosporidium*, *Giardia*, natural pollutants and beads. All subfigures share the same scale bar.108

Figure 4-10: Visualization on 2D latent space of traditional deep classification-based model and deep metric learning. (a) traditional deep classification-based model, (b) deep metric learning based model.109

Figure 4-11: Visualization on intermediate layers with t-SNE on deep metric learning and traditional classification-based model. (a, c, e) the lower, middle, and high level of deep metric learning, (b, d, f) the lower, middle, and high level of traditional deep classification-based model.111

Figure 4-12: Visualize of the GMM result with deep metric learning. Blue color is the natural pollutants and beads, the yellow color is the *Cryptosporidium* and the green color is the *Giardia*. The purple colour with white boundary is the estimated GMM cluster.112

Figure 5-1: Schematic of sieve analysis device that is divided of four sieves, which each sieve reduces open size to classifier the objects into bins.119

Figure 5-2: Large particles in dynamic light scattering. Time-dependent fluctuations in scattering intensity can be used to calculate size. Large particles move slow, so that the intensity of scattered light fluctuates slowly.....120

Figure 5-3: Small particle in dynamic light scattering. Time-dependent fluctuations in scattering intensity can be used to calculate size. Small particles move faster, so that the intensity of scattered light fluctuates more quickly.121

Figure 5-4: The schematic of the optical system for an imaging flow cytometry. The bioparticles passthrough the imaging centre and each channel records 2D images from a particularly CCD or CMOS camera simultaneously.....122

Figure 5-5: 2D images of different channel. Each channel recodes 2D image from particularly imaging modality. (a) bright-field image, (b) fluorescent image, (c) side scatter image.122

Figure 5-6: 2D CCD or CMOS camera. The 2D CCD/CMOS camera has a height H and a width W . Each pixel is a rectangle with a pixel size (PH and PW). Normally, PH equal to PW123

Figure 5-7: Different stage of size measurement pipeline. (a) Bioparticle image. (b) Segmentation. (c) Bioparticle Shape. (d) Bioparticle size in pixel numbers. (e) Bioparticle size in um . (a), (b) and (c) share the same scale bar.....125

Figure 5-8: Different processing stage of segmentation and pixel measurement: (a) Gaussian blur, (b) canny detector, (c) erode, (d) dilate, (e) find contours, and (f) estimate shape. All subfigures are in the same scale bar.126

Figure 5-9: A Gaussian kernel with $\sigma = 0.25$127

Figure 5-10: Horizontal and vertical direction kernels. (a) G_x and (b) G_y128

Figure 5-11: Dilate operation.....129

Figure 5-12: Erode operation.130

Figure 5-13: Graphic representation of Intersection over Union (IoU).....	134
Figure 5-14: Error analysis of segmentation algorithm. The image output of the segmentation algorithm of: (a) Segmentation results of <i>Giardia</i> , and (b) Segmentation results of <i>Cryptosporidium</i> . All subfigures are in the same scale bar.	134
Figure 5-15: Intersection over Union results of the segmentation algorithm.	135
Figure 5-16: Quadratic curve-based calibration.	136
Figure 5-17: The length distribution in both axes of microplastic particles within a distribution range. The circle represents the population distribution with the error in 2σ range. Red color ones are the physical diameters of beads, pink color ones are the beads sizes based on assumption, and cyan color ones are the beads sizes based on machine learning calibration.....	137
Figure 5-18: Measurement of individual microplastic particles sizes distribution: (a) 3 <i>um</i> , (b) 5 <i>um</i> , (c) 8 <i>um</i> , (d)10 <i>um</i> , (e)12 <i>um</i> , and (f)15 <i>um</i>	139
Figure 5-19: Measurement on <i>Cryptosporidium</i> and <i>Giardia</i> . (a) Length of the <i>Giardia</i> . (b) Width of the <i>Giardia</i> . (c) Diameter of the <i>Cryptosporidium</i>	141

List of Tables

Table 2-1: Summary on machine learning papers with non-commercial IIFC	50
Table 2-2: Summary on machine learning papers with commercial IIFC.....	51
Table 3-1: Network parameters	64
Table 3-2: IRBs search for base network.....	68
Table 3-3: Evaluation metrics.....	81
Table 3-4: Average accuracy, precision, recall and F1-score on test dataset	82
Table 3-5: Individual precision and recall score on multiple classification	86
Table 3-6: Error rate of different approaches on binary classification.....	88
Table 4-1: Network parameters of the embedding network	99
Table 4-2: Estimated parameter of GMM with deep metric learning.....	110
Table 4-3: Precision, recall and F1-score on test dataset.....	113
Table 4-4: Confusion matrix of traditional deep classification	114
Table 4-5: Confusion matrix of deep metric learning-based classification	114
Table 4-6: <i>Cryptosporidium</i> and <i>Giardia</i> detection using deep metric learning....	115
Table 5-1: Measurement error analysis	138
Table 5-2: Measurement results on bioparticles	140

Nomenclature

IFC	Imaging flow cytometry
DL	Deep learning
DNN	Deep neural network
CNN	Convolutional neural network
CCD	Charge-coupled device
CMOS	Complementary metal-oxide-semiconductor
PMT	Photo-multiplier tubes
APD	Avalanche photodiode
TDI	Time delay and integration
MIFC	Multi-field-of-view imaging flow cytometry
STEAM	Serial time-encoded amplified microscopy
FIRE	Fluorescence imaging using radiofrequency-ragged emission
ABD	Area-based diameter
ESD	Equivalent spherical diameter
HoG	Histogram of gradient
GB	Gradient boosting
RF	Random forest
KNN	K-Nearest neighbours
SVM	Support vector machine
ReLU	Rectifier linear unit
GD	Gradient descent
BGD	Batch gradient descent

SGD	Stochastic gradient descent
MBGD	Mini-batch gradient descent
ILSVRC	ImageNet large scale visual recognition challenge
SSD	Single shot multibox detector
R-MOD	Real-time moving object detector
FCNN	Fully convolutional neural network
VIFFI	Virtual-freezing fluorescence imaging
IRB	Inverted residual block
FC	Fully-connected
t-SNE	t-distributed stochastic neighbour embedding
FNR	False negative rate
FPR	False positive rate
EER	Equal error rate
FLOPS	Floating-point operations
GAP	Global average pooling
IoU	Intersection over union
RMSE	Root mean square error
RB	Residual network blocks
SQ	Stochastic quantization

Chapter 1 Introduction

1.1 Motivation

Flow cytometry is one of the most widely adopted technologies for bioparticle analysis in disease diagnostics [9-11], food inspection [12, 13], and water quality monitoring [14-16] among other bioparticle detection applications [17-21]. Conventional flow cytometry identifies bioparticles according to bioparticle optical signatures such as fluorescent profiles [17]. Bioparticle such as cells are often immunolabeled with fluorescent tags that target cell-specific biomarkers to facilitate analysis. Fluorescent labelling may alter cell properties and interfere with downstream analyses [22], and no knowledge of cell morphology may be derived from the information-sparse data obtained using conventional flow cytometry.

Morphology is an important characteristic feature of bioparticles. These images are packed with rich information which could be used for bioparticle classification [3, 23-29]. Gold standards primarily rely on bioparticles morphology with microscopy-based imaging techniques for microbial identification [30]. For instance, pathogenic protozoans in drinking water, *Cryptosporidium* and *Giardia* as shown in **Figure 1-1**, are detected by analysing the morphology in microscopy images [31]. However, the traditional microscopy-based imaging approach requires labour-intensive sample preparation and time-consuming manual data analysis, which significantly hinders its application in high-throughput bioparticles analysis [32].

Imaging-based flow cytometry is a high-throughput image acquisition

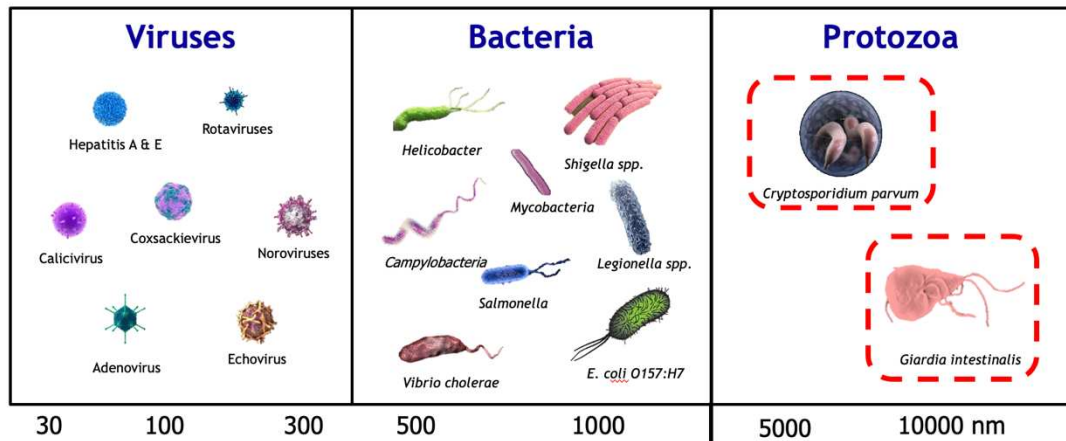


Figure 1-1: Categories of *Cryptosporidium*, *Giardia* and other viruses and bacteria.

technology by coupling imaging techniques with flow cytometry. Imaging flow cytometry has a significantly higher throughput than traditional microscopy-based imaging techniques and is capable of capturing thousands to millions of bioparticle images in a second using various imaging modalities [33]. The acquired images are packed with rich information on bioparticle morphologies. Nevertheless, the vast number of information-rich images imposes a great challenge for data analysis.

Machine learning is often used to identify bioparticles such as cells, planktons and microalgae during image analysis [34, 35]. Except for a few cases where machine learning is used for image correction, enhancement and reconstruction [27], a number of machine learning algorithms have been developed for imaging-based cell detection and classification [5, 25, 36-39] with reported high accuracy in these works, but they were relatively slow and required high computation resources beyond the capability of embedded systems in order to build affordable imaging flow cytometry machine. For instance, a highly accurate algorithm based on a sophisticated densely connected neural network [40] for high-throughput cancer cell detection in blood was reported [5]. It shows a good detection performance, but its speed was limited to 100 fps, and it relies on a high-performance platform with high-end Nvidia GeForce GTX 1080Ti GPU. Intelligent image-activated cell sorting 2.0 achieve a higher throughput cell

sorting up to 2000 events per seconds, but requires a 8-PC server with 8 multicore CPUs and GPUs (NVIDIA GeForce GTX 1080 Ti) for image processing with deep learning [4]. Most existing machine learning algorithms developed for imaging flow cytometry primarily focus on the detection accuracy but overlook the efficiency in terms of speed and hardware requirements, which has practical implications.

Deep learning with deep neural network has achieved surpassing performances for many applications with rich of data and supervised learning [5, 25, 36-39]. However, state-of-art deep neural networks still met challenges in practical applications such as the early warning system [8]. In that applications, the number of targets bioparticles are extremely small, and the number of background images are extremely large. Therefore, it requires the models to have a performance with a low false alarm as well as a high recover rate. Moreover, the target bioparticles are difficult to be collected for building representant training dataset. For example, the appearance of collected images of the bioparticles in the testing environments may be different from them in the training dataset in the particle applications. Those introduce difficulties to recognize them with traditional deep neural networks for the state-of-art deep neural networks assumed a static, closed world and cannot work well on open-set problem [41].

Some imaging systems offer an image-based size measurement as well as high-throughput bioparticle analysis [42]. Information-rich micrographs of individual particles are acquired at a high speed, and the particle size is determined by converting the pixel to length at a fixed conversion ratio that is calculated theoretically based on the specification of the optical components, called a *calibration*. For example, a single pixel corresponds to 0.33 μm in Amnis® ImageStream®X Mk II imaging flow cytometer [43] with a 60 \times objective, and 0.5

μm with a 40 \times objective according to the manufacturer's specification. However, it has been noticed that this fixed conversion ratio does not always lead to accurate particle sizing.

In summary, machine learning especially deep learning has spurred significant progress in high-throughput bioparticle images detection and sizing. Machine learning is a part of artificial intelligence and is capable of studying the experience from the training bioparticle data. It is widely adopted technologies for bioparticle analysis in disease diagnostics, food inspection, and water quality monitoring among other bioparticle detection applications. Current challenges in bioparticle images detection and sizing are high-throughput, achieving comparable accuracies using lower-cost embedded hardware, and high-accuracy sizing, etc.

1.2 Objectives

The major objective of this doctorate thesis is to develop machine learning algorithms for detecting and sizing of microscale bioparticle by using machine learning especially deep learning methods. Specifically, three types of areas are explored and discussed, namely: (1) Deep learning-enabled high-speed bioparticle detection; (2) Deep metric learning for rare bioparticle detection; (3) Machine learning-based pipeline for bioparticle sizing.

To enable low power and high throughput inference on resource limited embedded devices, for example low cost Nvidia Jetson TX2 [44], a deep learning-enabled real-time detection program for high-throughput and label-free bioparticle, such as *Cryptosporidium*, *Giardia*, microplastic and other pollutants in the water, is proposed. This intelligent detection platform combines imaging-flow cytometry and an efficient neural network developed known as MCellNet with the optimized target for high processing speed and similar detection accuracy.

For dealing with rare bioparticle detection problem, a deep neural network model based on metric learning is proposed. The algorithm leverages convolutional neural network to study the rich features inside training dataset and learns a distinct metric by using Siamese network and contrastive loss which learned to maximize the distance of different classes and minimize the distance of similar classes. The experimental results demonstrate that deep metric learning studies good features and high performance comparing with the traditional deep learning on rare bioparticle detection.

To enable high accuracy bioparticle size measurement, an bioparticle sizing algorithm pipeline is developed by employing computer vision and machine learning. The algorithm automatically segments particles from the 2D micrographs, estimates

the pixel size of the particles, and predicts the actual size from the pixel information by using a machine learning model from mass training data. Compared to the conventional approaches, the results show that our intelligent pipeline offers more accurate particle sizing by learning from the mass calibration data. This will greatly extend conventional particle sizing and has great potential in the field of environmental monitoring, biomedical diagnostical, and material characterization, etc.

1.3 Major Contributions

The major contributions of this PhD thesis are summarized as in the follows:

- 1) An efficient image classification models of deep learning are derived which enables high-throughput deep learning system for predicting *Cryptosporidium* and *Giardia* in drinking water. This system combines imaging flow cytometry and an efficient artificial neural network called MCellNet, which achieves a classification accuracy $> 99.6\%$. The system can detect *Cryptosporidium* and *Giardia* with a sensitivity of 97.37% and a specificity of 99.95%. The high-speed analysis reaches 346 frames per second, outperforming the state-of-the-art deep learning algorithm MobileNetV2 in speed (251 frames per second) with a comparable classification accuracy. The reported system empowers rapid, accurate, and high throughput bioparticle detection in clinical diagnostics, environmental monitoring and other potential biosensing applications.
- 2) A model of deep metric learning for inlier bioparticle detection is derived and the deep metric learning algorithm leverages convolutional neural network to study the rich feature inside the dataset and learns distinct metric by using a Siamese network and contrastive loss, which learns to maximize the distance between dissimilar classes and simultaneously minimizes the distance between similar class. Deep metric learning offers a generative and uses distance information to model the similarity of the images by learning a functional maps from images pixels to latent space, playing a vital role in rare object detection. Experiments show that the deep metric neural network achieves a high accuracy of 99.86% in classification, 98.89% in precision rate, 99.16% in recall rate and zero false alarm rate for inlier bioparticle detection.

The reported model leverages the use of the imaging flow cytometry in biomedical diagnosis, environmental monitoring, and other biosensing applications.

- 3) An intelligent pipeline for bioparticle sizing by employing computer vision and machine learning for accuracy sizing is built up and the intelligent pipeline automatically identifies and segments particles from the micrographs, measures the pixel size of the particles, and converts the pixel into actual size using a machine learning model learned from large amounts of training data. Compared to conventional approaches, the particle size determined by the machine learning model only has a mean percentage error of 4.2% which is five times better than the methods using a fixed pixel-to-size conversion ratio (23.3%). This method leverages the use of different intelligent imaging systems such as imaging flow cytometry for high accurate particle sizing and promises great potential in a wide range of applications in the field of environmental sensing, biomedical diagnostics, and material characterization.
- 4) A bright-field image database of *Cryptosporidium parvum* oocysts (2,082 images) and *Giardia lamblia* cyst (3,569 images), 1.54-um beads (3,466 images), 3-um beads (3,457 images), 4-um beads (5,783 images), 4.6-um beads (2,188 images), 5-um beads (9,637 images), 5.64-um beads (3,285 images), 8-um beads (3,066 images), 10-um (8,270 images), 12-um (4,704 images), 15-um beads (2,813 images), and natural pollutants of various shape and sizes (27,826 images) is built up. Each image is patched to a standard size with 120×120 pixels and it can be used to test bioparticle detection algorithms.

1.4 Organization of the Thesis

This thesis is organized with six Chapters. The introduction Chapter of this thesis covers the motivation, objective, major contribution and organization of this thesis as presented in this Chapter. The motivation section shows how this PhD research is important and why it was carried out. The objective section states the main focus of this thesis. The contribution section lists the important findings and innovations in both technological and theoretical point of view.

In Chapter two, a literature survey on current machine learning technologies in bioparticle imaging analysing is introduced. This Chapter provides an overview of basic knowledge of intelligent imaging flow cytometry, the evolution of machine learning and the typical applications, and how machine learning can be applied for assisted intelligent imaging flow cytometry. The future perspectives of machine learning in intelligent imaging flow cytometry are also discussed.

In Chapter three, a deep learning-enabled high-throughput system for predicting bioparticle is reported. This system combines imaging flow cytometry and an efficient artificial neural network called MCellNet. In order to evaluate the bioparticle classification models, a bioparticle images dataset is built up. That bright-field bioparticle image dataset includes *Cryptosporidium parvum* oocysts, *Giardia lamblia* cyst, beads and natural pollutants in the water. Moreover, a visualization analysing on the dataset is conducted. Furthermore, a deep learning-enabled high-throughput system for predicting *Cryptosporidium* and *Giardia* in drinking water is reported.

In Chapter four, a deep metric learning based model for detecting rare bioparticle is studied. The deep metric learning based model studies a distance function to measure the similarity between the samples and has better classification

accuracy than transitional deep learning model by applying similarity measure between samples with increasing the similarity distance between negative samples and reducing the similarity distance between positive samples.

In Chapter five, a bioparticle size measurement pipeline that combines image analysis and machine learning is studied for high accuracy measuring the bioparticle size on bioparticle imaging system. The pipeline consists of an image segmentation module for bioparticle identification and a machine learning model for accurate pixel-to-size conversion. The segmentation algorithm include traditional computer vision-based approaches with Gaussian pre-processing, Canny detector, dilation, erosion, and contours detection algorithms. The pixel-to-size conversion used a quadratic model for generating a prediction model for micrographs size measurement.

In Chapter six, a summary with the major contributions of this thesis and the recommend future works are presented.

Chapter 2 Literature Survey

Imaging flow cytometry is widely adopted in environmental monitoring, clinical diagnostics, precision agriculture and other potential biosensing applications. Intelligent imaging flow cytometry incorporates machine learning algorithms for automated image analysis with high throughput in measurement, processing, identification, and sorting of biological entities. Recently, machine learning, especially deep learning has taken over in the field of data analysis for imaging flow cytometry and promised fantastic performance in intelligent imaging flow cytometry. This Chapter provides an overview of basic knowledge of intelligent imaging flow cytometry, the evolution of machine learning and the typical applications, and how machine learning can be applied for assisted intelligent imaging flow cytometry. Future perspectives of machine learning in intelligent imaging flow cytometry is also discussed.

2.1 Introduction

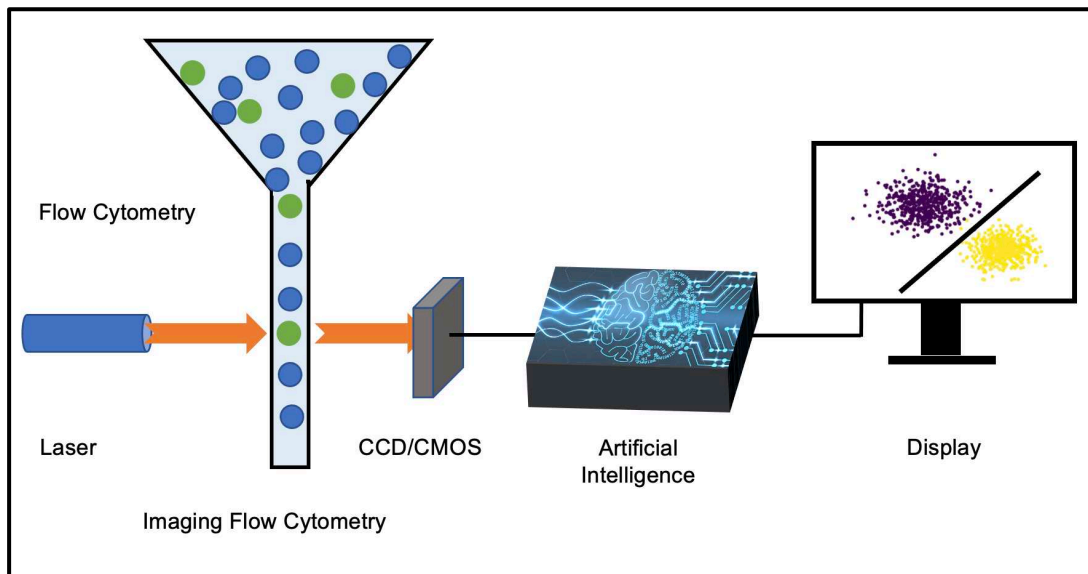


Figure 2-1: Overview of intelligent imaging flow cytometry.

Imaging flow cytometry is an analytical tool extensively used to detect, sort, count and measure phytoplankton, cells, and other bioparticles [17, 42, 45, 46]. By combining high-throughput flow cytometry with various imaging acquisition technologies such as multispectral imaging [33], imaging flow cytometry is capable of capturing thousands even millions of images with multiparametric morphology information, allowing automated high-throughput data collection. However, human expertise is often required for performing image analysis on traditional imaging flow cytometry.

Intelligent Imaging Flow Cytometry (IIFC) as shown in **Figure 2-1**, which combines imaging flow cytometry and machine intelligence, has been demonstrated for imaging-based high-throughput biosensing [3, 8, 23-29, 47]. Artificial intelligence (particularly deep neural network) plays a critical role in IIFC by providing new approaches of image enhancement, reconstruction, correction, and more importantly automated object recognition and identification of bioparticles and

other targets of interest. Advances in artificial intelligence lead the development of IIFC. Several IIFC using deep learning models, such as VGGNet, GoogleNet, ZooplanktoNet, DenseNets and deep active learning, have been demonstrated [4, 5, 40, 48]. A typical IIFC combines flow cytometry, image acquisition technologies (laser/image sensors), and artificial intelligence. It supports multiparametric analysis and high-throughput detection of the properties of bioparticle from hundreds to millions of particles per second. IIFC are widely used in clinical diagnostics [49], environmental monitoring [50], precision agriculture [51], and other potential biosensing applications [17-21].

In this Chapter, we focus on recent development in IIFC from the perspective of imaging technologies, the evolution of machine learning for computer vision, and machine learning techniques that are developed specifically for IIFC. The emergent imaging technologies such as multispectral imaging, multi-field-of-view imaging, and serial time-encoded amplified microscopy (STEAM) are discussed to reveal more distinctive features in the bioparticle images. To understand machine learning in imaging flow cytometry, we introduce the fundamentals of visual understanding, the evolution and knowledge of machine learning, increasing the understanding of machine learning in visual perception. Next, we review the most 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.

2.2 Machine Learning

2.2.1 Machine Vision and Image Analysis

Imaging flow cytometry technologies enable capturing and analysing the images of single bioparticles in high quality and high throughput. Besides the challenges in image acquisition, storage and processing, image analysis also requires significant efforts in the development of IIFC, which promotes the advancements in machine vision.

The working principle of machine vision system [52] 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 formed by the reflection of the object surface, mainly the visible and infrared parts, because this process is based on optical physics and solid-state physics, which can represent the real world.

2.2.2 Traditional Machine Learning

Since that time, a theoretical framework for object recognition was conceptualized, as well as several general vision theoretical frameworks, visual integration theoretical frameworks based on perceptual feature groups, 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 has greatly

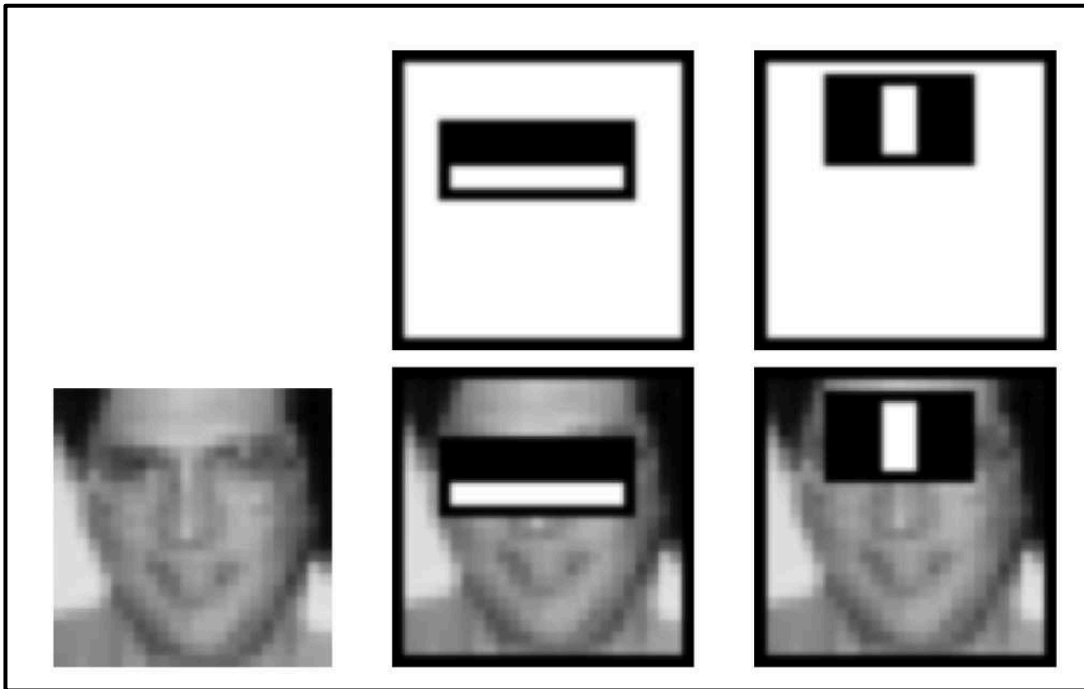


Figure 2-2: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlaid on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose. Reproduced with permission from IEEE [6].

has developed vigorously with emerging new concepts and theories.

Before the era of deep learning used in machine vision, image analysis methods could be roughly divided into the following five categories: image perception, image pre-processing, feature extraction, inference prediction, and recognition [53]. In the early stage of machine learning, among the dominant statistical machine learning groups, little attention was paid to features. The design feature is to combine these pixel values of the image in a statistical or non-statistical form to express the part or whole object that one wants to identify or detect.

As an illustration we show a face-recognition approach using in real-time using Haar-like features to locate a face as shown in **Figure 2-2** [6]. The Viola / Jones facial

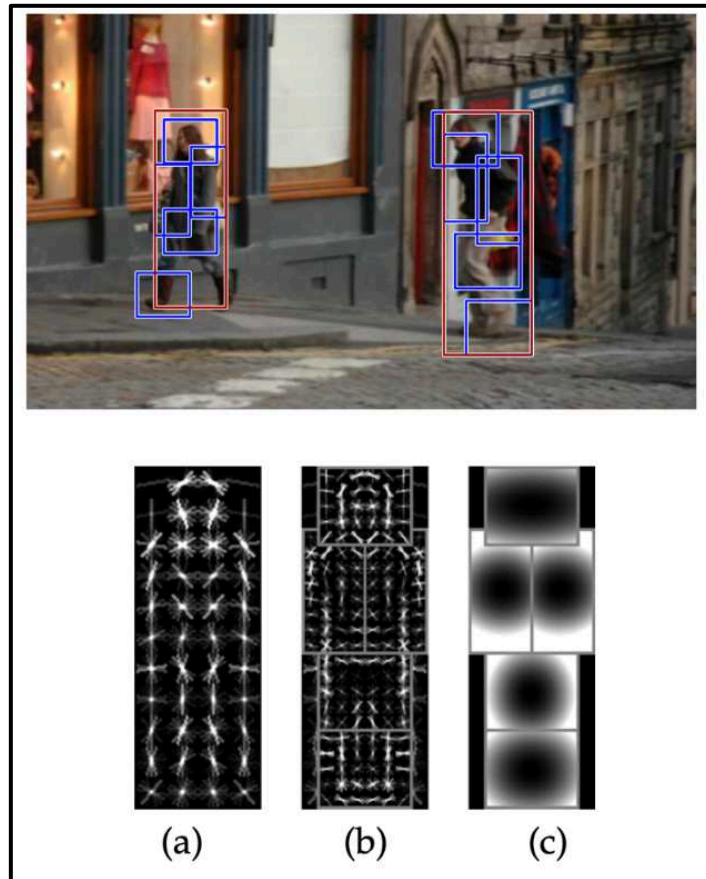


Figure 2-3: Detections obtained with a single component person model. The model is defined by a coarse root filter (a), several higher resolution part filters (b) and a spatial model for the location of each part relative to the root (c). The filters specify weights for histogram of oriented gradients features. Their visualization show the positive weights at different orientations. The visualization of the spatial models reflects the “cost” of placing the center of a part at different locations relative to the root. Reproduced with permission from IEEE [7].

detector is a powerful binary classifier consisting of several weak classifiers and is still widely studied today. However, it is time-consuming in the learning phase because adaptive boosting (Adaboost) is used to train the cascade of weak classifiers, such as finding the object of interest (*e.g.*, 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 it immediately.

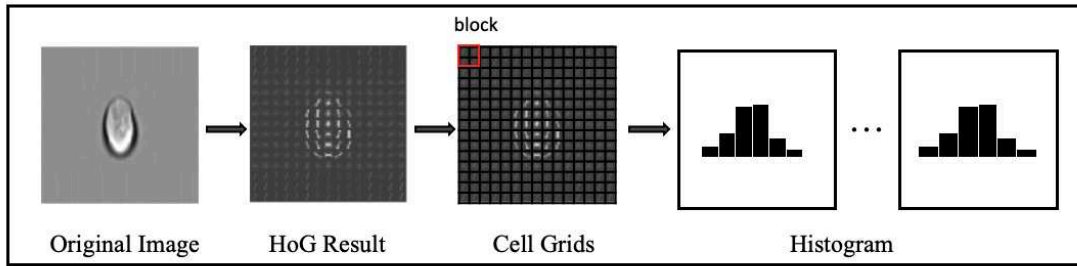


Figure 2-4: Working principle of histograms of oriented Gradients.

This whole process is repeated multiple times on various hierarchy of image scales.

In 2009, another important feature-based milestone work called Deformable Part Models (DPM) was developed [7] and is shown in **Figure 2-3**. DPM decomposes the object into partial collections, 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. 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) [54] feature 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 to 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.

HoG is a good feature descriptor that has been successfully deployed in human detection problem [55]. HoG has an advantage on densely capturing the gradient information of images, which is similar to SIFT [56], but it demands fewer computation resources. HoG is also resistant to the lighting conditions, *e.g.*, reduces shadows influence and other illumination variations such as smaller rotation and

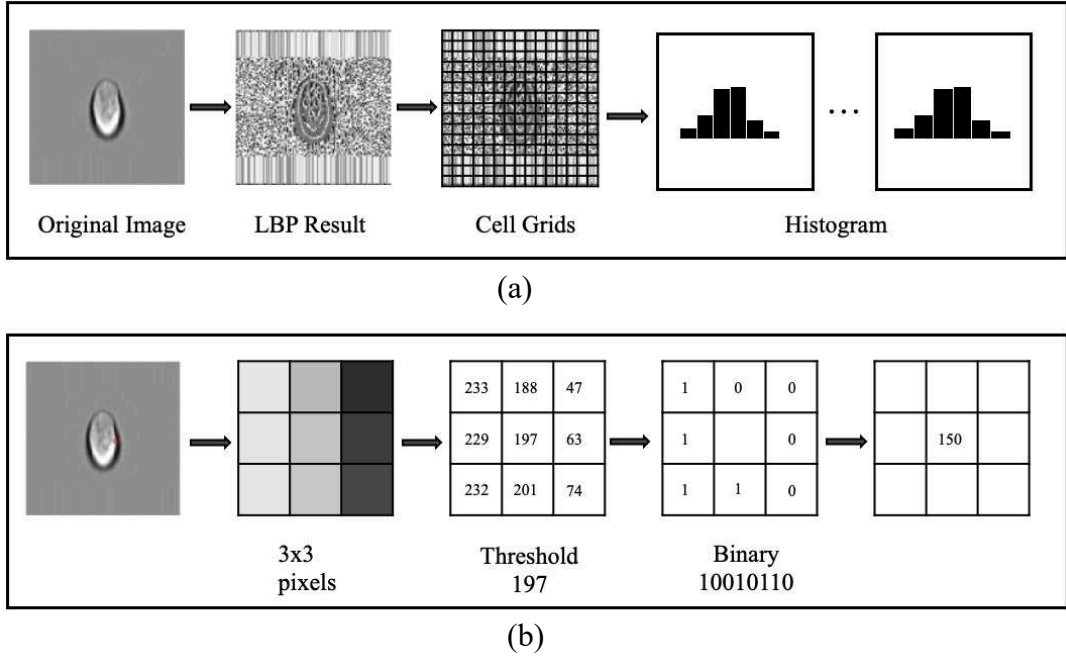


Figure 2-5: Local Binary Patterns. (a) The procedure diagram on local binary patterns histograms. (b) The diagram on how to calculate Local Binary Patterns.

translation with the gradient and histogram algorithms. As depicted in **Figure 2-4**, HoG calculates on small bioparticles in a window of 8×8 pixels. In that window, the direction of gradient and magnitude are calculated by

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

$$G_x = H_x * I(x, y), G_y = H_y * I(x, y) \quad (2-1b)$$

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

Finally, an unoriented gradients histogram of those 8×8 bioparticles are generated and put into 9 bins. Each bin is corresponded to the angles of direction of gradient in 0, 20, 40, 60, 80, 100, 120, 140, and 160 degrees. Since the gradient magnitude of the image is mostly sensitive to lighting, normalization on the histogram is desirable. The unoriented gradient histogram would guide to more robust feature sets because it can eliminate the effect of variations when the lighting conditions are varying.

Local Binary Patterns (LBP) is a popular texture feature with an excellent face detection performance [57, 58]. LBP excels in differentiating bright pixels from the dark background, which is used to describe edges, lines, spots, and flat areas. The procedure diagram of LBP features extraction is depicted in **Figure 2-5a**. First, the original input image is arranged into individual small bioparticles with 8×8 pixels. Then, LBP feature of each bioparticle is calculated by comparing the intensity of the 8 neighbouring pixels with that center pixel and generating an 8-bit binary number in which 0 or 1 indicate that the intensity of the neighbouring pixel is lesser or higher than the center pixel, respectively (**Figure 2-5b**).

Demonstrated examples of using traditional machine learning with imaging flow cytometry include an open-source imaging flow cytometry data analysis solution that uses the features generated from CellProfiler and together with Gradient Boosting (GB) [59] classifier and a random forest (RF) [60] classifier to recognize the Jurkat cells [61], and the identification of label-free white blood cells using the features generated from CellProfiler and classifiers such as K-Nearest Neighbours (KNN), AdaBoost, GB, RF, and Support Vector Machine (SVM) [62]. The SVM classifier [63] was one of the most popular discriminative classifier before the era of deep learning. It translates the vector of training data into a higher dimensional space and performs the discrimination. By doing this, the optimal hyper-plane can be generated, which splits the dataset into different classes via a training process. In the inference phase, it can be used to categorize new examples. SVMs can be expressed as the following optimization problem:

$$\begin{aligned}
 & \text{minimize } \frac{1}{2} W^T W + C \sum_{i=1}^n \xi_i \\
 & \text{subject to } y_i(W^T \varphi(x_i) + b) > 1 - \xi_i \\
 & \xi_i \geq 0, i = 1, \dots, n.
 \end{aligned} \tag{2-2}$$

where the two-class problem (binary problem) was defined as $y \in \{1, -1\}$. $W \in \mathcal{R}^D$ is the weight, ξ is the margin constant, b is the bias, and $C \in \mathcal{R}_+$ 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 it can be easier to generate the boundary of decision surfaces. A good choice for φ is to use Radial Basis Function kernel is used 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.

Distance metric such as the Mahalanobis distance classifier is an extension of the Least-Square multi-class maximum likelihood classifier taking cross-correlations into account [64, 65]. It measures the number of standard deviation distance d with calculated distance of x to a dataset and a mean u_i . The covariance matrix is defined as the equation Σ_i^{-1} and T is a standard transpose operation. The result class is predicted by measuring the distance from x to classes i and assuming the result has the minimal distance from the true predicted class. The Mahalanobis distance can be reduced to the Euclidean distance when the covariance matrix is the identity matrix. The equation of Mahalanobis distance is expressed as [65]

$$d(x, u) = (x - u_i)^T \Sigma_i^{-1} (x - u_i) \quad (2-3)$$

The classic problem of machine vision is 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., identify any object in unpredictable environment. The prior art can only solve well in the recognition of specific targets, such as simple geometric figure recognition [66], face recognition [67], printed or handwritten document recognition [68], and vehicle recognition [69]. Unfortunately, these recognition often needs to have a specific

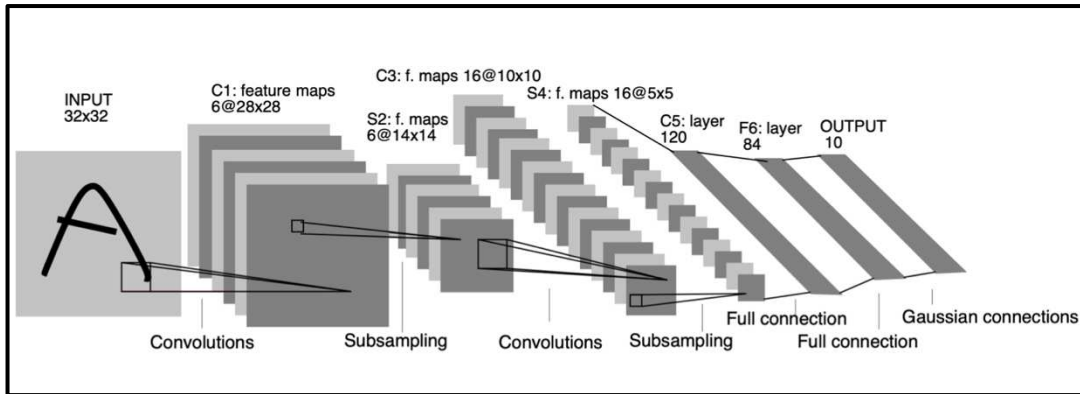


Figure 2-6: The architecture of LeNet-5 neural network. A convolutional neural network for digital recognition. Reproduced with permission from IEEE [2].

lighting, background, and target posture requirements in a specific environment. Designing features by hand requires a lot of experience such as a profound understanding of the field and data. It may also require a lot of debugging. Moreover, machine vision engineers not only need to manually design features, but also design a more suitable classifier algorithm for the problem. Meanwhile, 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.

2.2.3 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 designing 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 (CNN) has made this idea possible, and the research of computer vision based on deep learning has also developed rapidly. LeCun, a pioneer of the connectionist approach to AI proposed the first Convolutional Neural Network in LeNet [2] as shown in **Figure 2-6** in 1989. The input image is a 32×32 grayscale image. The first layer

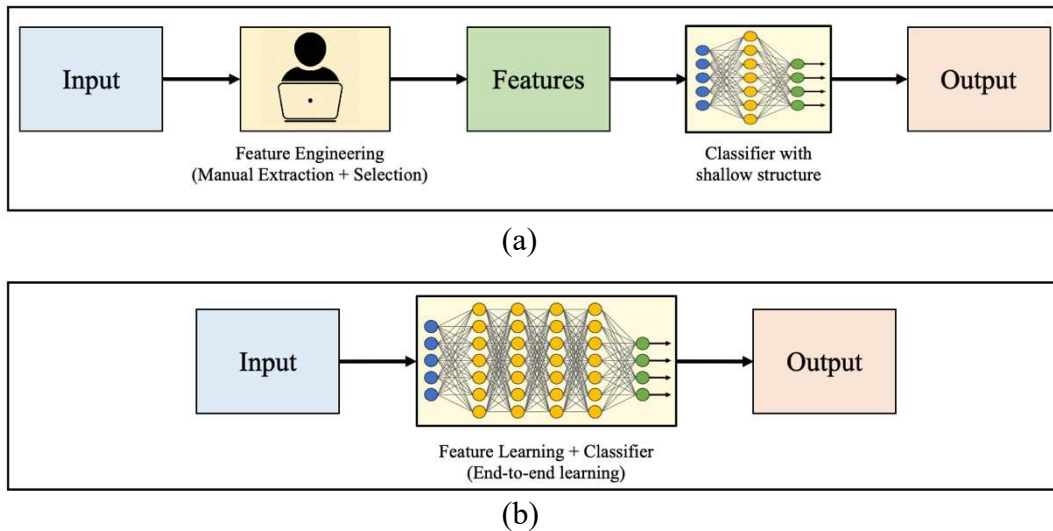


Figure 2-7: Comparison between (a) traditional handcrafted approach and (b) deep learning.

undergoes a set of convolution sums and generates six 28×28 feature maps, which pass a pooling layer to get six 14×14 feature maps and pass a convolution layer to generate sixteen 10×10 convolution layers. Next, they pass the pooling layer to generate sixteen 5×5 feature maps. It was used to classify hand-written digits 0-9 with two fully connected layers as 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 LSVRC [70]. Nowadays, deep learning has been applied to a variety of areas and big progress has been made in those fields, including visual recognition [71], speech recognition [72], biomedicine [73], and natural language processing [74], etc.

Deep learning methods are well-suited to constructing architectures that can be trained end-to-end from image data to result. This approach reduces manual engineering in the traditional approach as shown in **Figure 2-7**. It can automatically build multiple levels of representation of data with abstraction by end-to-end manner. For example, the first layer studies the edges or colors information. The second layer studies the motifs information. The third layer may learn the eyes and nose

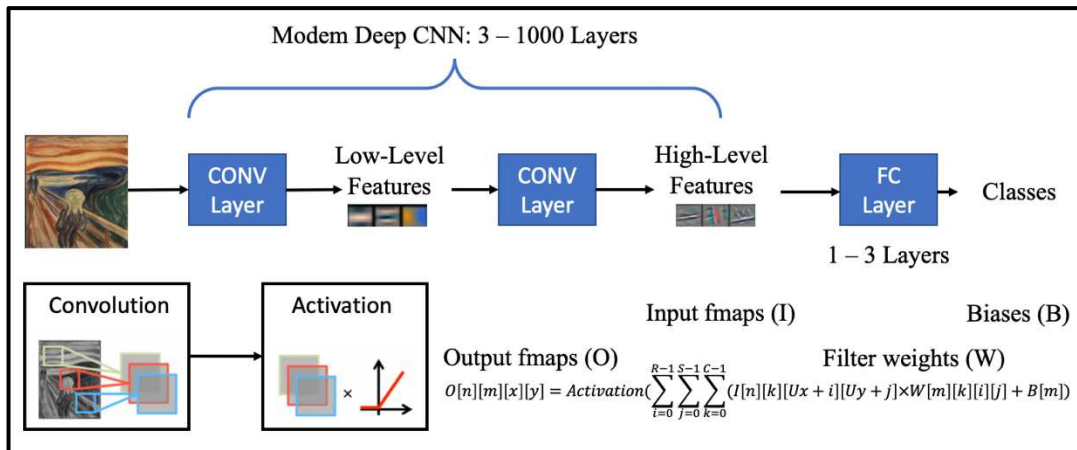


Figure 2-8: Convolutional neural network.

information. Finally, it 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 (*e.g.*, ReLU layer), and pooling layer (*e.g.*, max-pooling), fully-connected layer as shown in **Figure 2-8**.

The convolution function is used to extract the features from the input. The basic operation of convolution is illustrated in **Figure 2-9a**. On the left side of the figure, the input has a dimension of $N \times N \times D$. It convolutes with a kernel H with a size of $k \times k \times D$. Finally, a feature with $N \times N \times H$ dimension 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, H number of kernels will generate H layers of output. Unique positions will be generated from left to right and top to bottom without pad with $N = 32$ and $k = 3$, $30 \times 30 \left((N - k + 1) \times (N - k + 1) \right)$.

The ReLU layer, as shown in **Figure 2-9b**, is a rectified linear activation function. It implements a non-linear “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. Comparing with other non-linear functions

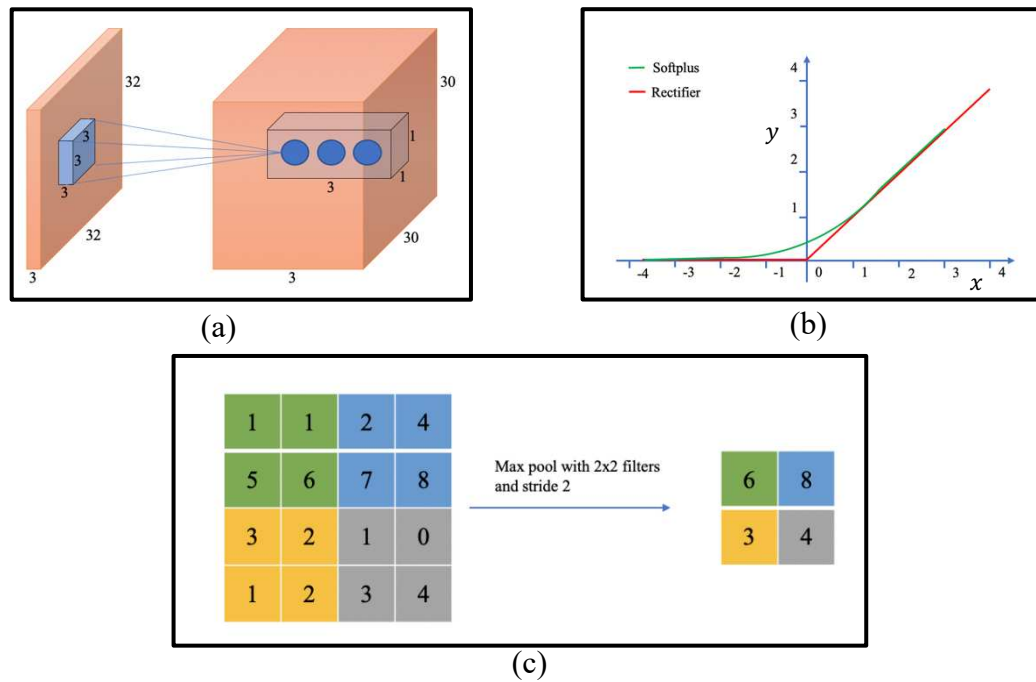


Figure 2-9: Layers of convolutional neural network. (a) Convolutional operation. (b) Rectified Linear Unit (ReLU). (c) Max-pooling operation.

such as sigmoid, hyperbolic tangent, and absolute of hyperbolic tangent, the networks with ReLU learn several folds faster than equivalents of other non-linear functions. The max-pooling layer as depicted in **Figure 2-9c** 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 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 2.

One or several fully-connected layers are normally added to the last layer of a convolutional neural network and acts as a classifier for the final decision. It always takes a vector of m input as the input volume and generates n output with the function, which is expressed as

$$Y_n = W X_m + b \quad (2-4)$$

where m is the input dimension, which is computed with the weights matrix 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 convolution layers and the weights in dense layers. The parameters are optimized by the backpropagation algorithm, which uses a gradient descent method to optimize the model iteratively by minimizing a loss function (*e.g.*, cross-entropy loss). The three frequently used GD algorithms are batch gradient descent, stochastic gradient descent, and mini-batch gradient descent. Softmax regression [75] of the classification layer outputs was employed to train the network., which can be written as

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

During the training, the loss is calculated from the model input with forwarding propagation whereby the loss difference backwards 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 stabilized after the iterative process.

A convolutional neural network is a powerful learning-ability neural network that is widely used by image classification and segmentation. The convolutional neural network is inspired by the natural visual perception mechanism from the human's perception system. The early attempt was the proposed neocognitron system [76] in 1980. By improving the neocognitron, LeCun proposed LeNet-5 to solve the handwritten digits, which established the modern framework of the CNN [2]. LeNet-5 gave a basic idea of CNN such as convolutional neural networks use a three-tier architecture: convolution, down sampling, and nonlinear activation functions. It extracts image space features by using convolution and reduces image average sparsity with down sampling. The activation function takes a hyperbolic tangent or

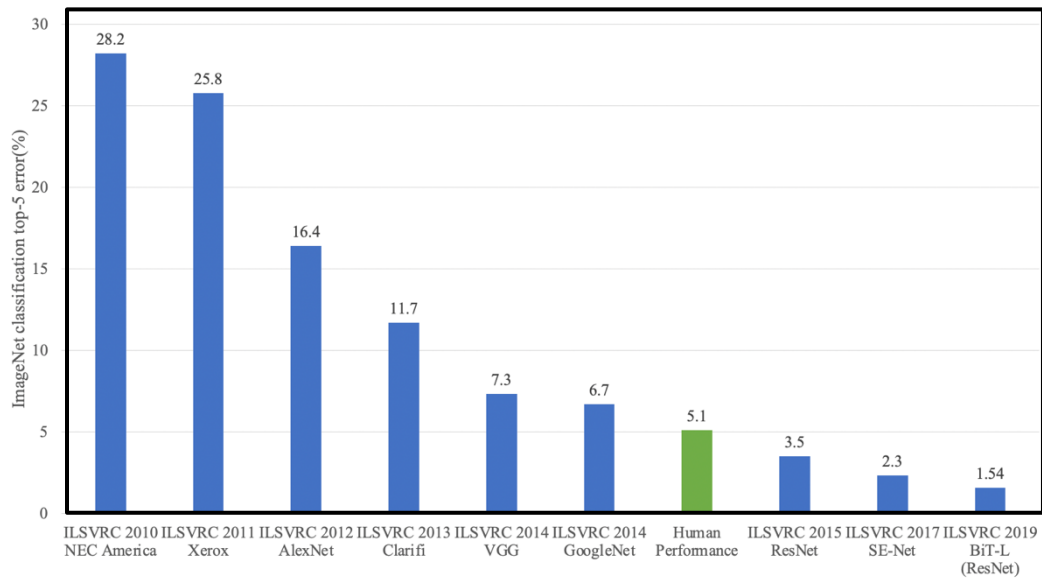


Figure 2-10: ImageNet challenge.

sigmoid function. 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 CNN to recognize the object directly from the original pixels with minimal pre-processing. However, due to the lack of large-scale training data and the computing power was limited, LeNet-5 could not work well on complex problems. From 1998 to 2010, the development of neural networks was intense in the machine-learning community, but not very visible to the computer vision community. The combination of theoretical advances or not only neural architectures but also optimisation methods (stochastic gradient descent, Nesterov accelerated descent, etc. [77]) with progress mostly with GPUs but also low-power CPUs, and fast, low latency disks such as SSDs brings cost-effective hardware to the world, making deep neural network computation affordable and opens the door for deep learning. In 2010, a GPU neural network was published [78].

In 2012, AlexNet was published [70], which is relatively deeper than LeNet's

network and won the first champion of the 2012 ImageNet competition as shown in **Figure 2-10**. AlexNet not only has deeper neural networks, but also learns more complex image with high-dimensional features. AlexNet introduced the ReLU function as an activation function. This activation function is convex and has no vanishing gradient for positive weights, considerably reducing computation time in the learning phase. Furthermore, it used the dropout technique to clip certain neurons during training to avoid over-fitting. It also introduced max-pooling technology and significantly reduce training time with GPU. After the success of AlexNet, researchers proposed other improvements, such as VGGNet [79], GoogleNet [80], Residual Network (ResNet) [81], MobileNetV2 [82], SENet [83], and BiT [84], etc.

From the structural point of view, one of the CNN's development direction 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 a better characterization. However, this also increases the overall complexity of the network and makes the network difficult. It can also easily overfit. As the depth of the network increases, the accuracy of the network should increase simultaneously. Adding layers significantly affects the parameter updating on the gradient propagating from the back to front. Increasing the network depth runs the risk of vanishing gradients, resulting in networks that are hard to optimize. In addition, the optimization problem becomes more difficult when the network becomes deeper, with larger parameter space. Therefore, simply increasing the network depth will result in higher training error. For example, the performance of a 56-layer network is not as good as that of the 20-layer network. Due to this, ResNet was designed with a residual module that allows

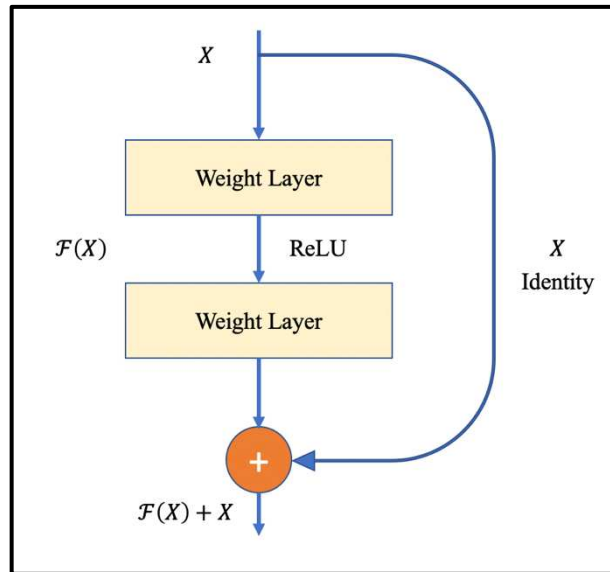


Figure 2-11: Residual learning building block.

us to train deeper networks [81].

The core idea of ResNet is to add a direct connection channel to the network, known as identity shortcut connection, which is the idea of Highway Network [85]. The network structure of traditional deep learning is a nonlinear transformation that is performed on the input while ResNet allows the original input information to be passed directly to the subsequent layers as shown in **Figure 2-11**. In Ref [81], the authors argued that stacking layers will not degrade the network performance with stacking identity mappings (a layer that does not do anything) upon the current network and the resulting architecture would perform the same. In such a case, the deeper model should not produce a higher training error than its shallower counterparts. They hypothesized that fitting the stacked layers to a residual mapping is easier than fitting them directly to the desired underlying mapping. 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. The comparison of VGGNet and ResNet is shown in **Figure 2-12**. The biggest difference of ResNet is the use of bypass connection to directly connect the input to the subsequent layers, which is also called shortcut or skip connections.

Most existing deep learning algorithms developed mainly focus on the detection accuracy but overlook the tradeoff between speed and hardware requirements. The use of high-end GPU system makes it possible to train complex and deep neural networks, but it is a major hurdle for mass deployment of these deep learning algorithms to commercial IFC for bioparticle analysis due to its high cost and high-power consumption. Very recently, faster and efficient deep learning models, such as MobileNet [86], DenseNet [87], SENet [83], and MobileNetV2 [82], have attracted great interest from the research community because they are able to achieve only slightly degraded classification accuracies in return for using lower-cost embedded hardware. Various methods have been proposed to improve network performance with those problems in various aspects. The recent improvements of CNN include the convolutional layer, pooling layer, activation function, loss function, regularization, optimization, and fast computing techniques.

MobileNet was proposed by Google LLC in 2017. It is a lightweight convolutional neural network focused on mobile devices and embedded devices. SENet won the 2017 ImageNet LSVRC (ImageNet last session) image classification task championship. Normal convolutional neural network aggregates spatial information and feature dimension (channels) information on a local receptive field in order to obtain global information. Many existing studies have shown the benefits

of enhancing spatial information dependence. However, SENet hopes to solve that by learning the importance of different channel characteristics and pay more attention to the dependence between channels. The characteristic of this model is that it proposes a new architecture "Squeeze-and-Excitation" (SE) block to explicitly model the dependencies between channels. Compared with the traditional CNN, it greatly reduces the model parameters and the amount of calculation under the premise of a small decrease in accuracy.

DenseNet is proposed in the CVPR2017 best paper "Densely Connected Convolutional Networks". It does not improve network performance from the perspective of increasing the depth and width of the network like the previous network but draws on the idea of ResNet's shortcut connections and establishes dense connections between each layer and all other layers behind. It can alleviate the vanishing gradient problem, strengthen feature propagation, encourage feature reuse, and greatly reduce the number of parameters. However, they need extensive optimizations for specific tasks.

MobileNetV2 (Inverted Residuals and Linear Bottlenecks) is an improved version based on the MobileNetV1 model, which has better accuracy and a smaller model than MobileNetV1. MobileNetV2 is characterized by the inverted residual with linear bottleneck. In the network, features can be further embedded into a low-dimensional subspace through transformation (for example, dimensionality reduction through 1×1 convolution). In addition, MobileNetV1 also reduces the dimensionality of the feature layer through the width factor α , which is a compromise between the amount of calculation and accuracy. However, since the layers of the deep convolutional neural network have a non-linear activation function, reducing the dimensionality of the activation space may cause a lot of information to be lost when

using ReLU for activation transformation in a low-dimensional space. Because in the training process, some units of the feature map will inevitably output values below 0. After these values pass through the activation function, they have no contribution to the subsequent features. Moreover, the convolution parameters is lost. The author demonstrates that the information is lost when dimension equals to 2 or 3 through the change process of the information of interest in. More important, when the dimension is between 15 and 30, there are significantly more information recovered.

Through the study of the activation function, the bottleneck in ResNet has been substantially improved for the problem of not bringing too much information loss when extracting features in low-dimensional space. One of the improvements is to use linear bottleneck (*e.g.*, remove ReLU) to do linear transformation instead of the original nonlinear activation transformation. In addition to the above operations, the author considers that in order to reduce the loss of information, the input feature map can be expanded first, and useful information can be embedded into the high-dimensional feature map as much as possible, and then proceed depth separable convolution and ReLU operations.

The Inverted Residual Block (IRB) in the MobileNetV2 includes a 1×1 expansion convolutional layer, a depthwise convolution layer, and a 1×1 projection. The depthwise convolution layer and 1×1 projection layer are referred as the depthwise separable convolution adopted by Xception [88], which consists of the depthwise convolution and follows by a pointwise convolution. The depthwise convolution is represented as

$$\widehat{X}_{x,y,z}^k = \delta(\sum_{i,j} \widehat{F}_{i,j,z}^k \cdot X_{x+i-1,y+j-1,z}^{k-1} + b_z^k) \quad (2-6)$$

where \widehat{F}^k is the depthwise filter in which the z_{th} channel in \widehat{F}^k only calculates with the z_{th} channel of X^{k-1} and produces the feature \widehat{X}^k in the z_{th} channel.

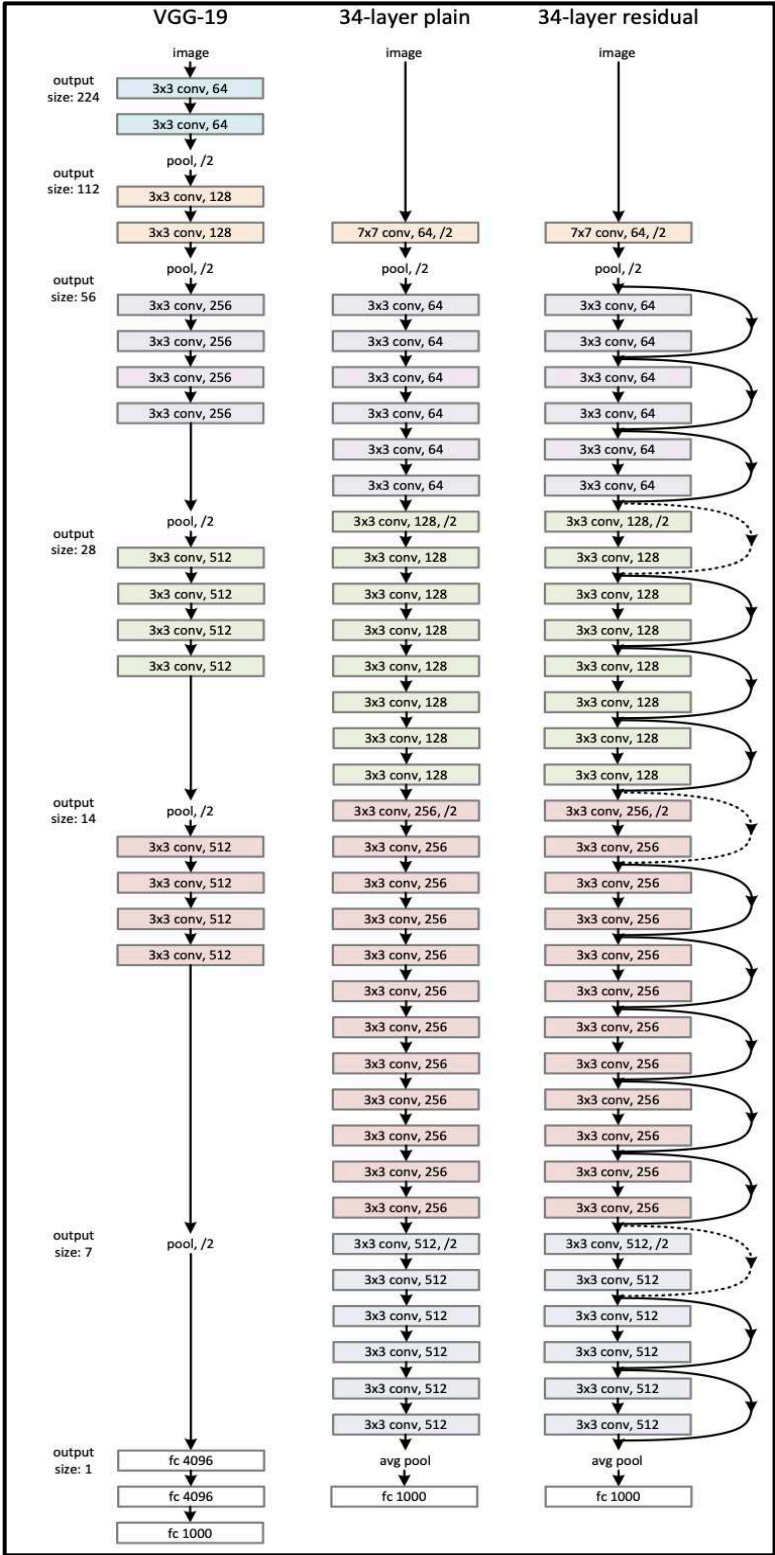


Figure 2-12: Examples of network architecture for ImageNet. the VGG-19 model is in the left, a plain network with 34 parameter layers is in the middle, and a residual network with 34 parameter layers in the right. The dotted shortcuts increase dimensions. Reproduced with permission from IEEE [81].

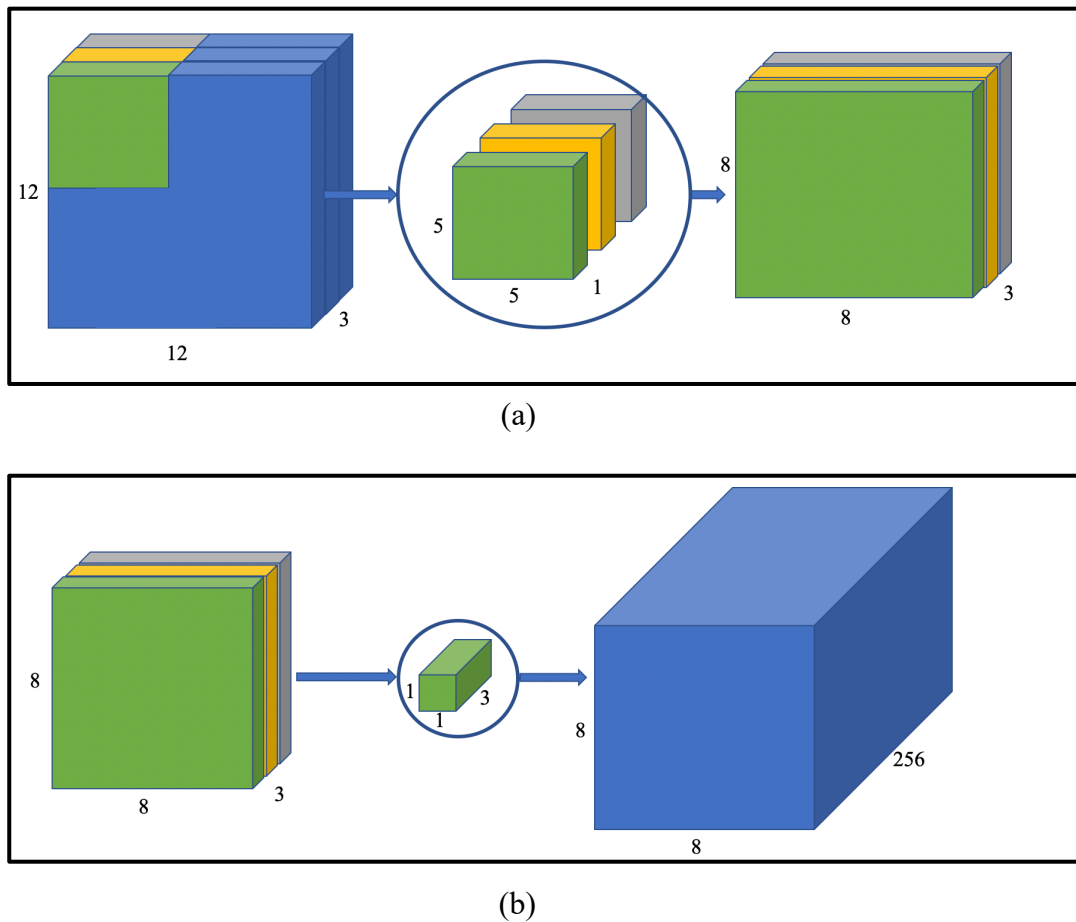


Figure 2-13: (a) Depthwise convolution. (b) Pointwise convolution with 256 kernels.

The depthwise separable convolution [86] 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 2-13a**. A pointwise convolution uses a 1×1 filter to produce the final activation map as depicted in **Figure 2-13b**. Compared to the traditional convolution, the computation reduction of the depthwise separable convolution is $\frac{1}{N} + \frac{1}{D_k^2}$, where N is the number of output channels, and D_k is the kernel size. Furthermore, IRB also increases the resource efficiency with its unique architecture. Besides, the skip connection structure is introduced to IRB, which allows the network to access features in earlier

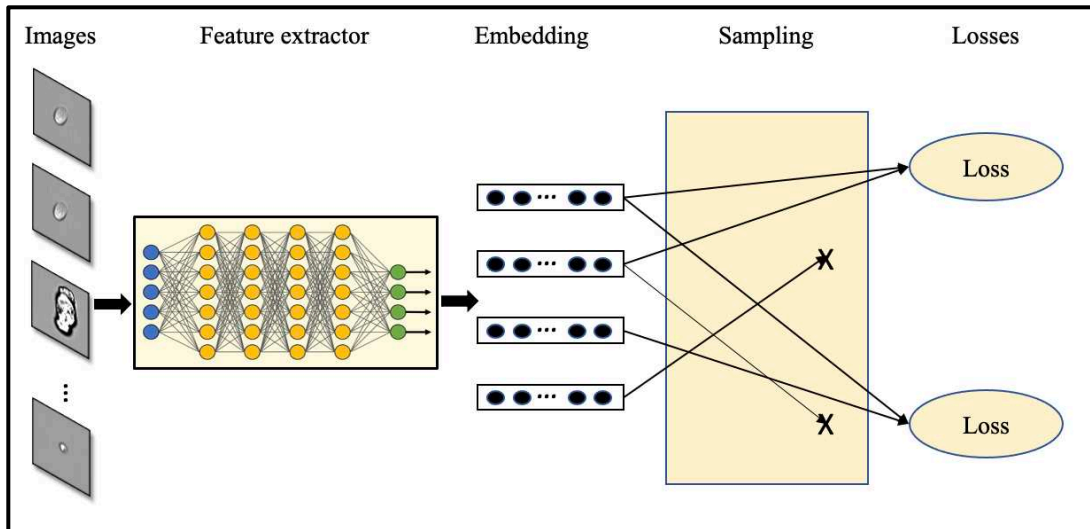


Figure 2-14: General pipeline of deep metric learning. It includes embedding network for extracting features to latent embedding space, sampling strategies, and loss functions.

stages and lead 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 one. Deep metric learning (DML) currently mainly uses the deep learning based embedding network to extract embedding, and then uses L2-distance to measure the distance in the embedding space [89]. In general, DML consists of three parts: feature extraction network to map embedding, a sampling strategy to combine the samples in a mini-batch into many sub-sets, and finally the loss function calculates the loss on each sub-set as shown in **Figure 2-14**. Such as deep metric learning with contrastive loss [90]. The sampling algorithm random selects an example as an anchor. Then random select another one image from rest images of same class as the positive pair and select one image from others class as the negative pair. The popular loss functions are contrastive loss [90], triplet loss [91], etc.

The contrastive loss is used to train Siamese network. 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 negative pair if

they are dissimilar. The contrastive loss is expressed as [90]

$$L\left(\{W^{(m)}, b^{(m)}\}_{m=1}^M\right) \quad (2-7)$$

$$= \sum_{(i,j) \in \mathcal{S}} h(d_f(\mathbf{x}_i, \mathbf{x}_j) - \tau_1) + \sum_{(i,j) \in \mathcal{D}} h(\tau_2 - d_f(\mathbf{x}_i, \mathbf{x}_j))$$

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

$$d_f(\mathbf{x}, \mathbf{y}) = \|f(\mathbf{x}) - f(\mathbf{y})\|_2$$

$$= \sqrt{(f(\mathbf{x}) - f(\mathbf{y}))^T (f(\mathbf{x}) - f(\mathbf{y}))} \quad (2-8)$$

where $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. The triplet loss is expressed as [91]

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

where x_a and x_p are from same class and x_n is from difference class.

2.3 Imaging Flow Cytometry

Technologies to obtain images with both high temporal and spatial resolution are critical but challenging [33]. The fundamental trade-off in imaging technologies is sensitivity, acquisition speed, and the amount of acquired information. Commonly, there are two types of detectors used for imaging: (1) multi-pixelated imaging devices (camera-based), such as charge-coupled device (CCD) and complementary metal-oxide-semiconductor (CMOS) [92], and (2) single-pixel photodetectors, *e.g.*, photomultiplier tube (PMT) and avalanche photodiode (APD) [93].

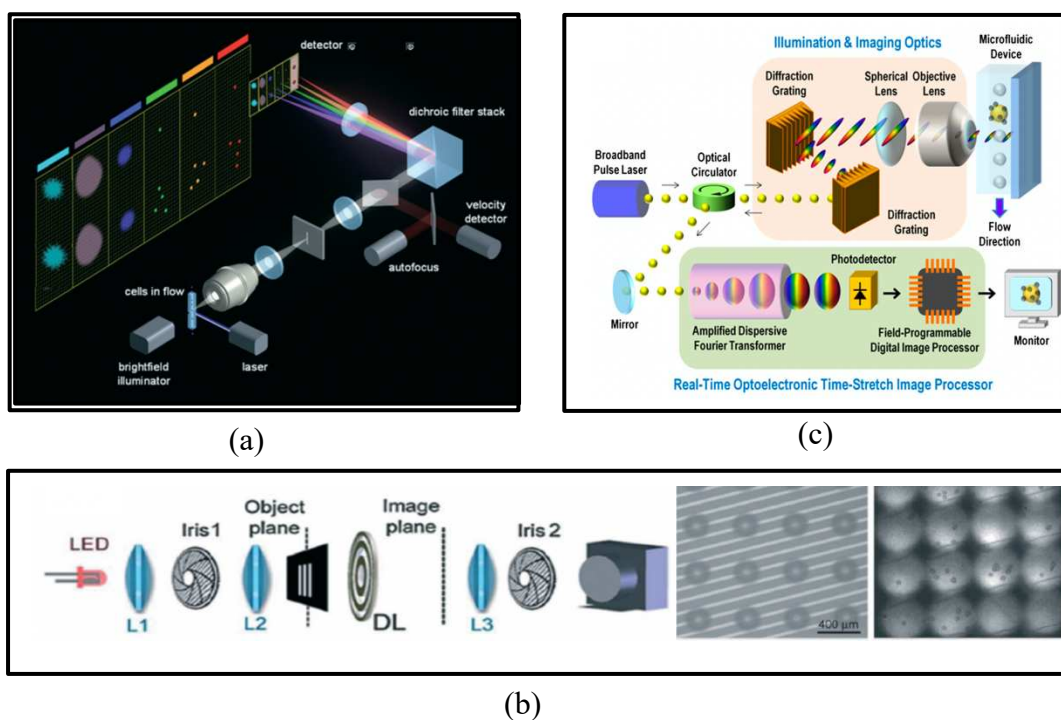


Figure 2-15: Optical systems of typical imaging flow cytometry. (a) Optical configuration of ImageStream imaging flow cytometry. (b) Multiple field-of-view imaging flow cytometer. (c) Schematic illustration of STEAM flow analyzer. Reproduced with permission from Royal Society of Chemistry [33].

The camera-based imaging flow cytometry has a dense 2D array of CCD or CMOS sensors, such as the commercial system ImageStream (**Figure 2-15a**) and FlowSight both developed by Millipore. They support multispectral imaging acquisition up to 12 images per bioparticle and three different imaging modes: brightfield, scattering, and fluorescence based on the time delay and integration (TDI) techniques [94-96]. The TDI sensor includes multiple rows of CCD or CMOS sensors. When applied to imaging, the objects move along the column direction and the imaging data are shifted row by row. It can read out weak imaging signal without motion blur even with increasing exposure time. Because the movement of the objects needs to be precisely synchronized with the vertical transfer direction on the rows of CCD or CMOS, the TDI technique requires an accurate control on the microfluidic

system and the circuit to ensure that bioparticles flow at a constant speed without rotation. Unfortunately, data transfer between the rows without gain (*e.g.*, electron multiplication) also limits it up to 3,000 bioparticles per second.

To increase the throughput, a multi-field-of-view imaging flow cytometry [97] was developed as depicted in **Figure 2-15b**. This method projects multiple fields of view into a 2D camera, such as microfabricating multiple parallel microfluidic channels with $N \times M$ micro-lens arrays to capture the image of the $N \times M$ channels 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 [98] is a technique used to avoid motion blur, which employs a pseudo-random code modulated excitation pulse to illuminate the object. A de-blurred algorithm is also applied with a known point spread function to reconstruct the final object.

PMT sensors provide superb sensitivity for photon signals with 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 the ultrafast STEAM [99, 100] as illustrated in **Figure 2-15c**. 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 grating for scanning and illuminating the bioparticle. The reflected signal is collected and stretched by a dispersive medium. Eventually, the rainbow signal is collected by an APD in series. STEAM can achieve a throughput of 100,000 bioparticles per second. Other examples using PMTs include fluorescence imaging by radiofrequency-tagged emission [101, 102] for high-speed fluorescence imaging, spatial-temporal

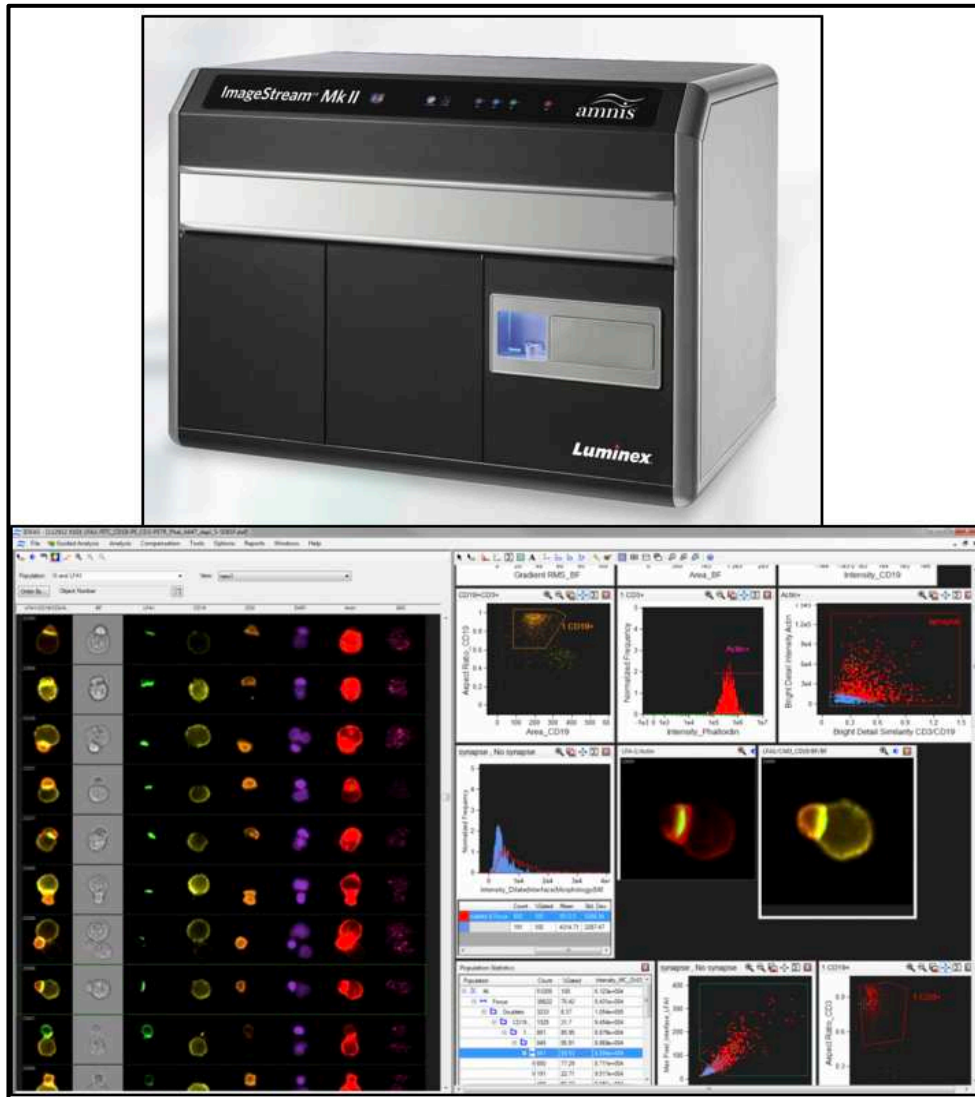


Figure 2-16: ImageStreamX Mk II.

transformation cytometry, etc.

The emerging commercial imaging flow cytometry empowers high-speed bioparticle splitting and microscopic imaging. For example, ImageStreamX Mk II as shown in **Figure 2-16** uses high-resolution and high-sensitivity objective lenses to produce brightfield, darkfield, fluorescence, and fluorescence intensity images [96]. It contributes significantly to the advancement of a wide range of quantitative, statistically robust cellular analysis and makes more insight into the following areas such as cellular classification, cellular processes, cell-to-cell interactions, microalgae morphology, population dynamics, etc.



Figure 2-17: FlowCam System

FlowCam as shown in **Figure 2-17** is another imaging flow cytometer that was originally developed by Fluid Imaging Technologies (Yarmouth, ME, the USA) to study the oceanic plankton [103]. It uses 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 single grayscale or color image of single bioparticles instantly. The software supports different features extraction such as area, area-based diameter, length, width, equivalent spherical diameter, and others properties [104].

Submersible Imaging FlowCytobot as shown in **Figure 2-18** is another type of imaging flow cytometer that can be submerged in water up to 40-m depth with a run for 6 months [105]. It can also transmit acquired data to the cloud in real-time. The Submersible Imaging FlowCytobot works similarly to a standard flow cytometer that



Figure 2-18: Submersible Imaging FlowCytobot

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 analyse large cells within $150\ \mu\text{m}$. An automated taxonomic classification is used to classify them into 22 categories for the comparative evaluation.

2.4 Deep Learning Applications

Intelligence imaging flow cytometry that combines imaging flow cytometry and machine intelligence has emerged as a promising platform for imaging-based high-throughput biosensing [106, 107]. With the development of imaging techniques, the imaging system could reflect the rich sets of bioparticle information that allow insightful and more rigorous analysis based on fluorescence signal through immunolabelling and scattering information originated from the interaction of the bioparticle structures with light. Based on the spatial arrangement of the bioparticle images, the analysis problem can be split into two classes: (1) images which contain multiple bioparticles, and (2) images that contain only a single bioparticles.

When one image contains multiple bioparticles, detection and tracking

problems occur. The detection task localizes all objects in the image with a bounding box such as applying deep learning to detect mitosis on breast history images. As the evolution with deep learning, two-stage model such as Faster-RCNN are used to detect malaria cells [108], and one-stage model such as SSD to detect neural cell [109]. Comparing with the previous model, Faster-RCNN and SSD provide better performance on the speed and detection accuracy. Nowadays, segmentation and detection joint approaches such as DeepLab [110] and Mask-RCNN [111] excel with the advantage of multi-task learning. DeepLab uses atrous convolution, atrous spatial pyramid pooling and probabilistic graphical models to achieve the qualitatively and quantitatively performance. On the other hand, Mask-RCNN provides a simple and flexible general framework for object instance segmentation.

The detection and tracking task, for example, a pipelined real-time imaging processing algorithm based on convolutional neural network is demonstrated for imaging flow cytometry [3]. This algorithm uses a simple convolution neural network to identify microbeads and cells. The real-time image processing pipeline uses FCNN, the Hungarian assignment algorithm and convolutional classification algorithm for the segmentation, tracking and classification the cells in the microfluidic based image flow cytometry. A microfluidic channel is monitored by a CMOS camera via a microscope. Then, the real-time image is processed by real-time moving object detector (R-MOD) system as depicted in **Figure 2-19**.

The R-MOD system contains two parts: (1) multiple objects tracking and (2) single-cell image acquisition and identification. The multiple objects tracking algorithm composes of three parts: image segmentation, detection, and tracking. The image segmentation algorithm is implied with a convolutional neural network, which performs a regression operation to convert the grayscale microscopy image to a

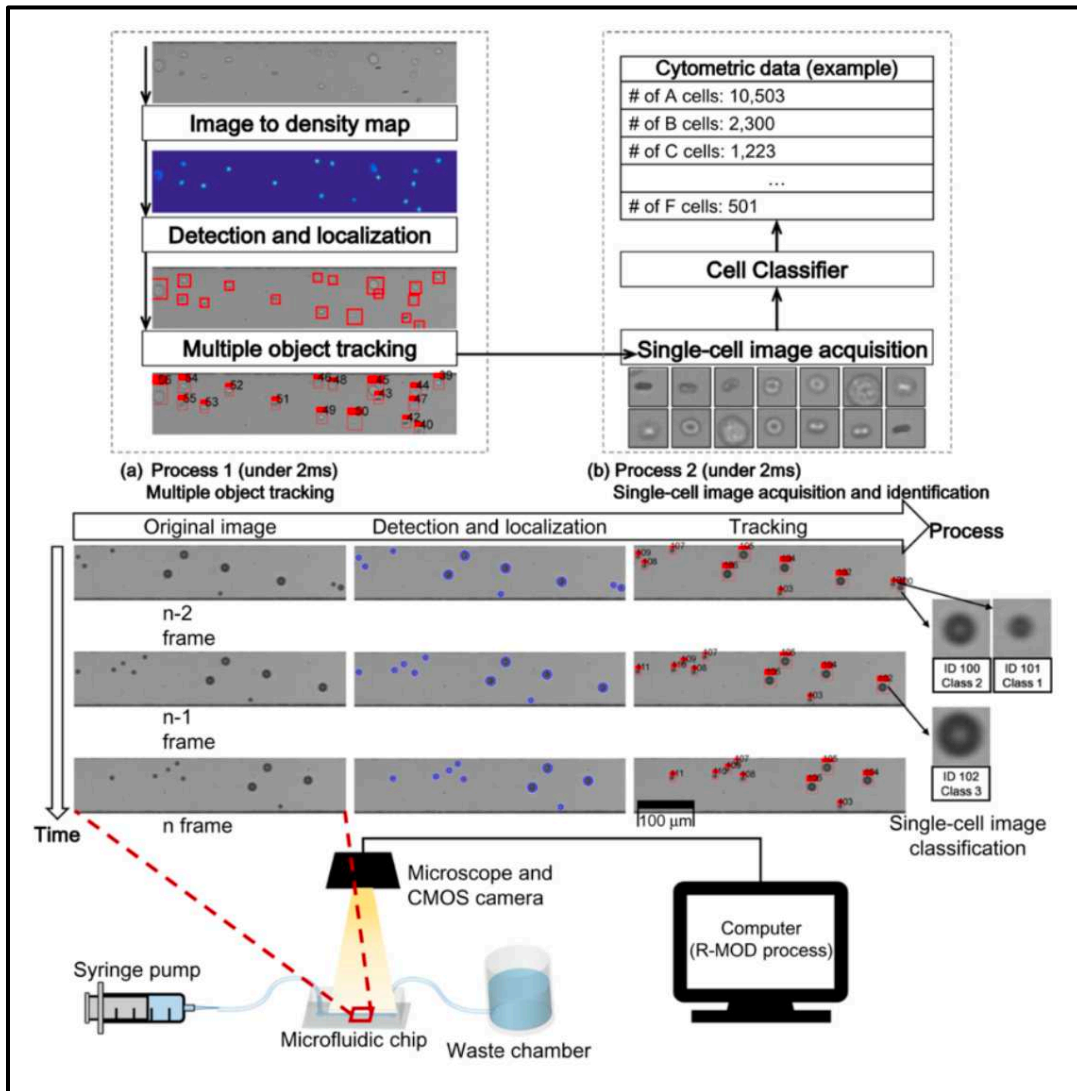


Figure 2-19: R-MOD (Real-time Moving Object Detector) system. Reproduced with permission from Nature Research [3].

probability density map. The detection algorithm finds the mean u and variance σ of the cell object. First, it finds the maximum pixel in the density map as the mean u . Then, the area around the maximum pixel with σ is cropped from the density map and the equation of σ is expressed as

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

This process continues until all gaussian distributions in the density map have been removed. The tracking algorithm uses the Hungarian algorithm to detect the objects in the consecutive images and assigns the object number to the detected

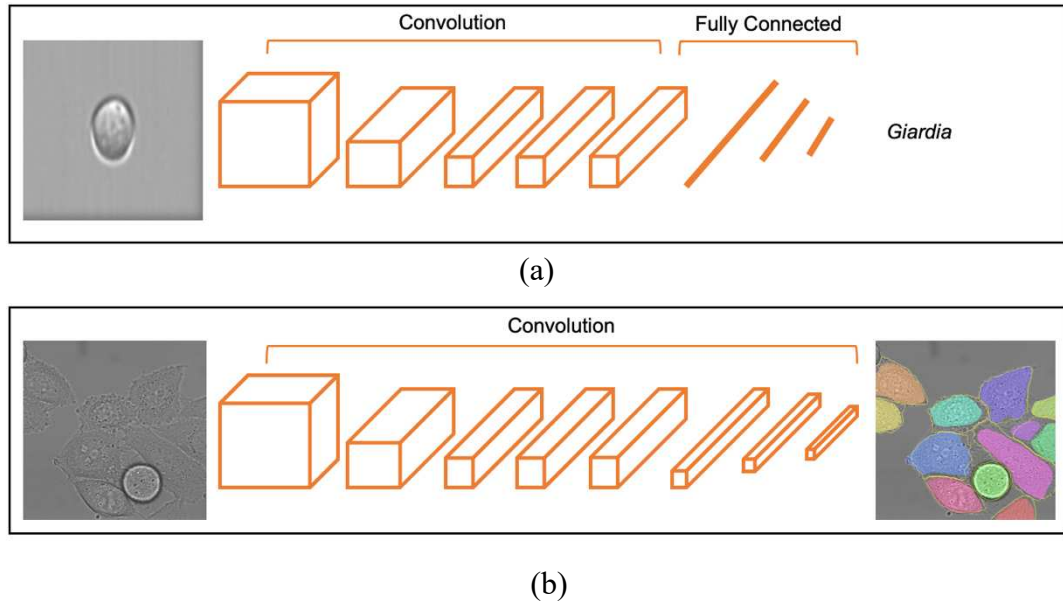


Figure 2-20: Classification and Segmentation. (a) Classification, (b) Segmentation.

About 21×21 pixels are cropped around the centre of the object. A classifiers-based CNN is used to determine the cell type and the experiments are performed with DI water and the dataset is not sufficiently complex. They also do not account of the natural containment particles in the water. Furthermore, the computation power of this algorithm is high and needs to be run on high-end GPU workstation. Future consideration on object detection and tracking algorithms for imaging flow cytometry include Recurrent YOLO [112], SiamMask [113], Deep SORT [114], Tracking R-CNN [115], etc.

Classification and segmentation are two fundamental problems in computer vision as well as single cell image analysis. The image classification (**Figure 2-20a**) predicts a label for a giving image, and the segmentation (**Figure 2-20b**) splits the digital imaging into subparts or super-pixels such that every super pixel has the same meaningful label. In general, the segmentation is a subset of 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.

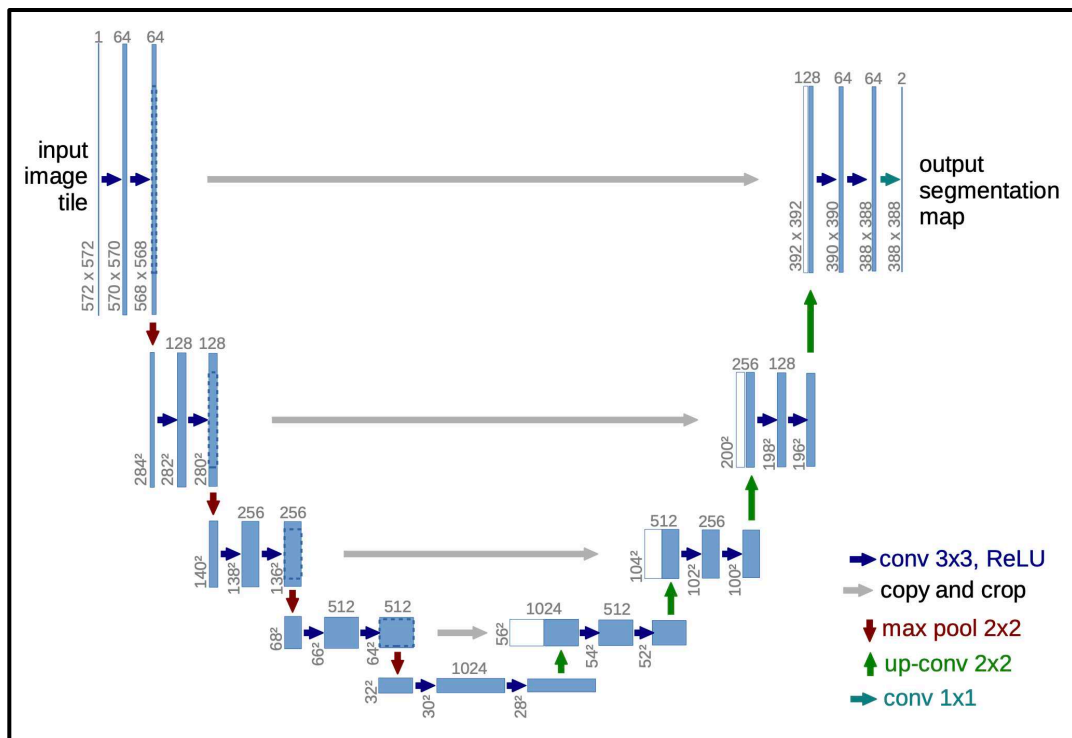


Figure 2-21: U-Net. Reproduced with permission from Nature Research [1].

Early attempt used fully convolutional neural networks (FCNNs) on segmentation and classified cell such as the HEP-2 cell specimen [116]. FCNN [117] were a type of state-of-art segmentation architecture on image segmentation task in 2014 and were trained to classify the HEP-2 cell specimen into seven catalogues. Comparing with the classification network, FCNN replaces the fully connected layers to 1×1 convolutional layers. The classification network generates a label for each image, but the FCNN gives a pixel label for every pixel. It learns a function, which maps the input pixel to the output pixel label. FCNN and CNN are different because the last three layers in the CNN network are one-dimensional vectors. The calculation method no longer involves convolution, and, therefore, the two-dimensional information is lost. While in the FCNNs, all three layers are converted into 1×1 convolution kernel with the equivalent vector length corresponding to the multi-

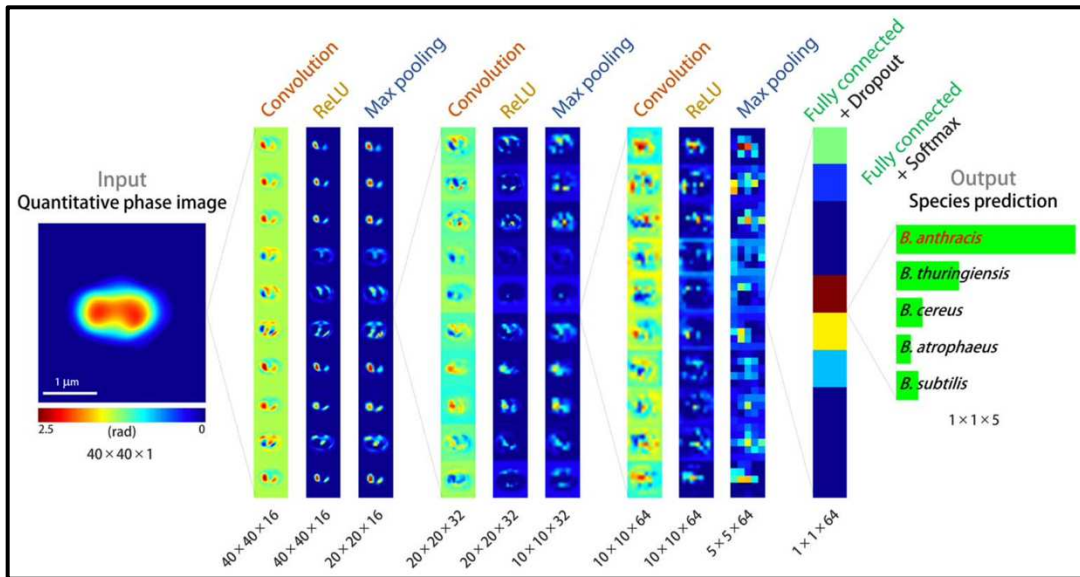


Figure 2-22: Holographic deep learning. Reproduced with permission from American Association for the Advancement of Science [8].

channel convolutional layer and the latter three layers in FCNNs are all convolution calculations. In the whole model, all layers are convolutional layers and without fully connected layers, which is called Full Convolution.

Recently, U-Net and its variations have dominated cell level segmentation and counting [1]. U-Net is a generic fully convolutional neural network 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 2-21** illustrates 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 convolutional neural network. 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 conventional layers. To increase the local information, information from the contracting path is combined with concatenation. At the end of the whole network, a

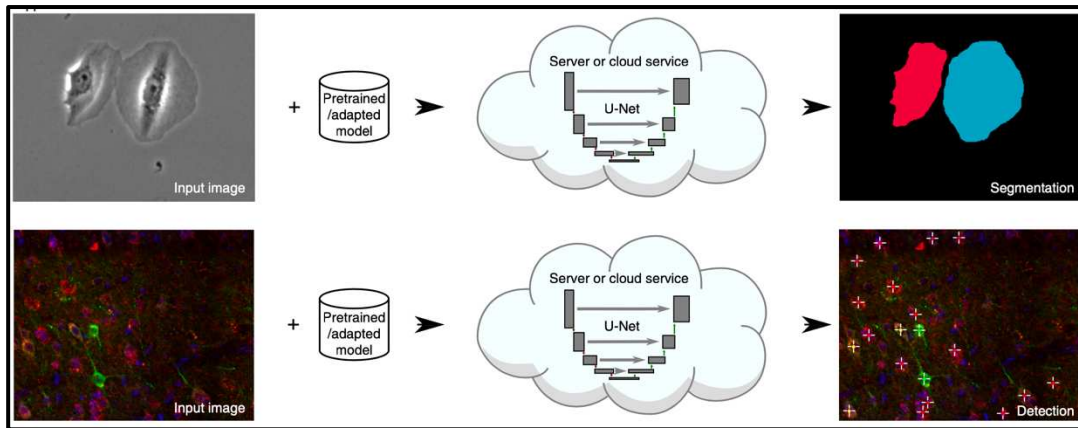


Figure 2-23: U-Net: deep learning for cell counting, detection, and morphometry. Reproduced from ref. [1] with permission from Nature Research.

1×1 convolutional layer is applied to map the feature map from 64 to 2 classes (cell and membrane).

Examples of using deep learning in bioparticle classification and detection on non-commercial IIFC and commercial IIFC are summarized in the **Table 2-1** and **Table 2-2** respectively. One of the examples is a deep convolutional neural network used as an early warning system for anthrax detection [8]. As shown in **Figure 2-22**, a dataset was built with difference anthrax samples such as *B. anthracis*, *B. thuringiensis*, *B. cereus*, *B. atrophaeus*, and *B. subtilis*. A deep convolutional neural network, HoloConvNet, was also built with convolutional neural network for anthrax classification. It has three convolutional layers, two fully connected layers, and achieves an accuracy of 96.3%.

An ImageJ plugin interface to U-Net is shown in **Figure 2-23** to enable non-machine-learning experts to count, segment, and detect cells [1]. The U-Net with ImageJ interface, which offers a step-by-step protocol for cell detection, counting, and segmentation such as the prediction of the center of the cell and the delineation of the outlines of individual cells. It can achieve results comparable with the human experts' level. For the training, it requires a relatively low number of annotated

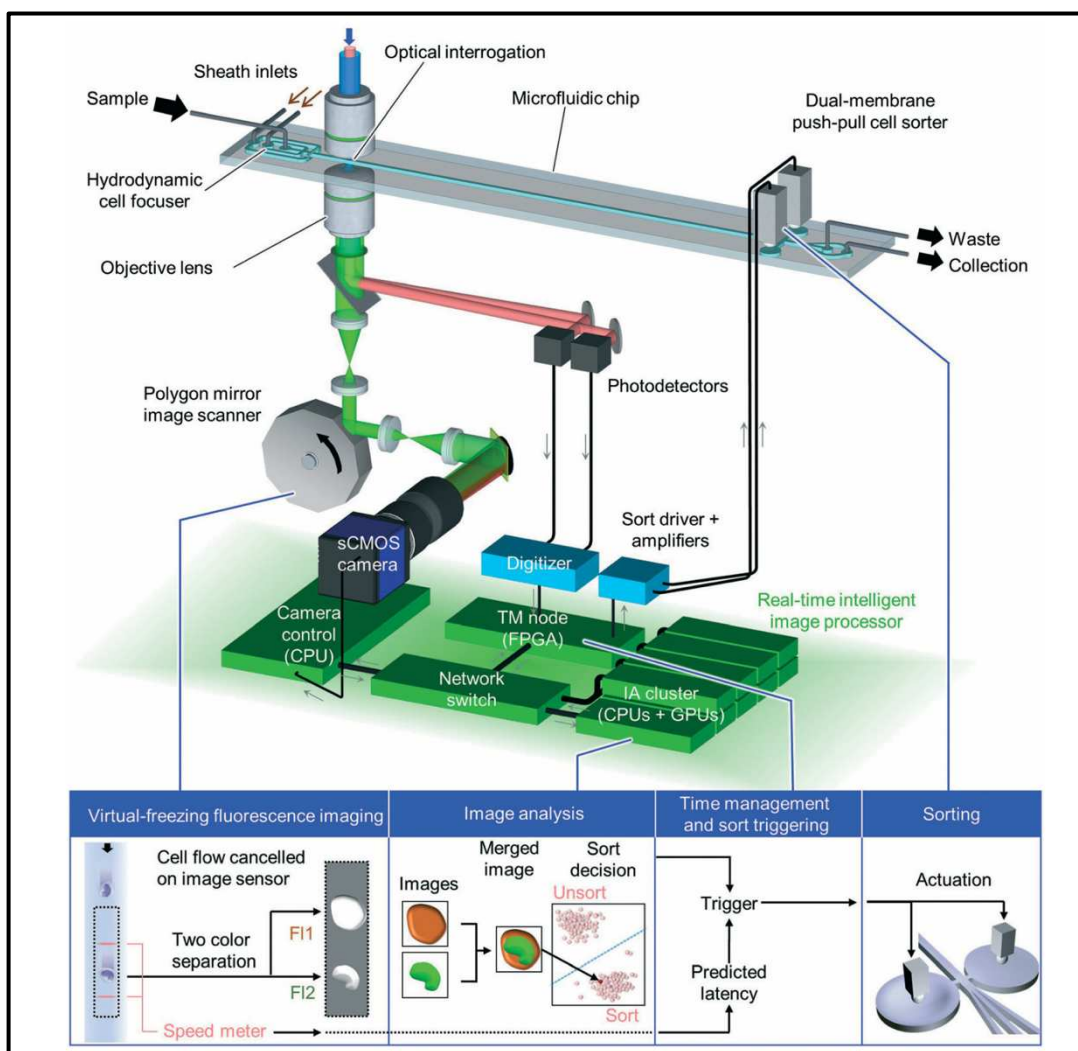


Figure 2-24: Intelligent Image-Activated Cell Sorting. Reproduced with permission from Royal Society of Chemistry [4].

images with special data augmentation, which needs more data for training on particularly difficult cases. Furthermore, it can run with a larger image size without any modification. Being 100% convolutional, it can deal with arbitrary image resolution.

A deep convolution neural network-enabled image-activated real-time cell sorter [4] with a processing speed of 2000 events per second was also demonstrated as depicted in **Figure 2-24**. However, it requires a complex hybrid hardware/soft data management system with 8-PC server with 8 multi-core CPUs and GPUs (NVIDIA GeForce GTX 1080 Ti) for image processing. First, the suspended cells in a sample

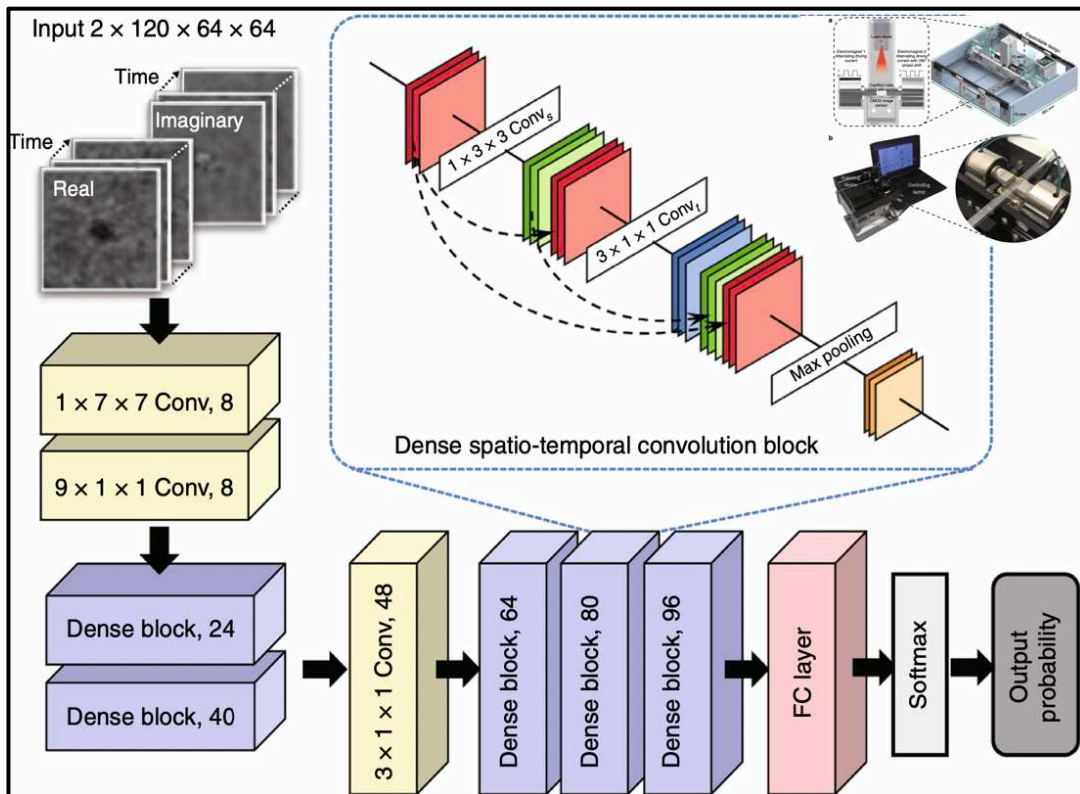


Figure 2-25: A classification network based on densely connected neural network. Reproduced from ref. [5] with permission from Nature Research.

tube are injected into the intelligent IACS, which are hydrodynamically focused to a single stream. Then, the cells are imaged by virtual-freezing fluorescence imaging [118]. Next, the images are analysed by a real-time intelligent image processor. Finally, the cells are sorted by a dual-membrane, which receives decisions from the image processor and employs cell push-pull mechanism for sorting. The whole process is operated in an automated and real-time manner.

Another example of a deep learning-enabled portable imaging flow cytometer was developed by Göröcs et al [27]. This device combined holographic imaging and neural network for the rapid detection of harmful algae in the water. The resolution of the device was limited to $25 \mu\text{m}$ and only suitable to identify large organisms. The highly accurate algorithm based on a sophisticated densely connected neural network [40] is shown in **Figure 2-25** for high-throughput cancer cell detection in blood [5].

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 in three dimensions (3D). Next, it detects those cells through a densely connected pseudo-3D convolutional neural network. It automatically detects those rare cells of interest based on the spatial-temporal features under a controlled magnetic force. It has a good sensing performance but its speed is limited to 100 fps, and it had to rely on a high-performance platform with Nvidia GeForce GTX 1080Ti GPU. High-end GPU empowers the training of complex deep neural networks, but it is a major hurdle for mass deployment of these machine learning algorithms to commercial imaging flow cytometry for real-time object identification and classification due to its high cost and high-power consumption.

Machine learning algorithms are used in emerging commercial imaging flow cytometry for phytoplankton analysis, label-free cell-cycle identification, white blood cell identification, or label-free leukemia monitoring, etc. In early days, traditional feature extraction and classifiers [61, 62, 119], such as AdaBoost, Lsboosting, GB, KNN, RF and SVM were used. With the emerging of deep learning, researchers move their interest to deep learning-based algorithms [120-123]. For examples, the conventional convolutional neural network, AlexNet [70], VGG [79], GoogleNet [80], ResNet [81], DenseNet [40], NasNet [124], and PyramidNet [125], etc. Among these deep learning algorithms, they are optimized to achieve high prediction accuracies and required high computational resources, such as Nvidia GTX 1080Ti, which are power hungry, expensive, and occupy huge footprint when designing IIFC.

Table 2-1: Summary on machine learning papers with non-commercial IIFC

Application	Imaging	Application	Dataset	Algorithm	Result	Hardware	Reference
Cell Classification	Time-stretch imaging	Classify the cells. THP1, MCF7, MB231, PBMC	Collected	CNN model	99 % accuracy	Nvidia Tesla K40c	[36]
Cell Sorting		Cell sorting	Collected	CNN model	NA	Nvidia GTX-1080Ti	[4]
Label-free Analysis of Natural Water Samples	Lens-free holographic imaging	Phase-contrast color images reconstruction and identification of plankton	Collected	CNN model	NA	NA	[27]
Label-Free Bioaerosol Sensing	Frequency-division-multiplexed (FDM) microscope	Label-Free Bioaerosol detection	Collected	Resnet based	accuracy >94%	NA	[25]
Real-time Inference and Cell Sorting	Time-stretch imaging	Cell sorting and cancer cell classification	Collected	CNN model	95% accuracy	Nvidia Tesla P100 GPU and Nvidia K80 GPU	[28]
Image Construction and Classification	Lens-less time-resolved holographic speckle imaging		Collected	CNN model		Nvidia GeForce GTX 1080Ti GPU	[5]
Label-free Detection of Cellular Drug Responses	Time-stretch imaging	Drug-treated and -untreated cells classification	Collected	linear SVM/CNN model	accuracy of 92%	NA	[24]
Label-free Cell Cycle Analysis	Microscopy	Label-free cell-cycle classification of Jurkat cells	Collected	LSBoosting	70.2±2.2% (G1), 90.1±1.1% (S), 96.8±0.3% (G2) and 44.0±8.4% (M)	NA	[23]
Parasites Detection	Lens-less time-resolved holographic speckle imaging	Parasitic detection	Collected	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	[38]
Label-free Detection	Microscopy	Detection Beads	Collected	CNN model	93.3% mAP	Nvidia GTX 1080	[3]
Reconstructing Cell Cycle and Disease Progression	Microscopy	Cell cycle of Jurkat cells/diabetic retinopathy	Collected	CNN model	accuracy of 98.73%	GPU	[126]

Table 2-2: Summary on machine learning papers with commercial IIFC

Application	Imaging	Application	Dataset	Algorithm	Result	Hardware	Reference
Imaging Analysis	ImageStream or FlowSight (Amnis)	Label-free cell-cycle classification of Jurkat cells/segment image and extract features/multiclass machine learning	Collected	Gradient Boosting Random Forest	92% accuracy	NA	[61]
Plankton Classification	NA	Plankton classification	WHOI, ZooScan, Kaggle	AlexNet, GoogleNet, VGG, Resnet, DenseNet	F1 <95%	Nvidia TitanX GPU	[121]
Microalgae Classification	FlowCAM	Microalgae classification	Collected	CNN model	88.59% of accuracy	Nvidia Titan X Pascal	[120]
Plankton and Coral Classification	NA	Automated system for monitoring underwater ecosystems	3 plankton and 2 coral datasets	AlexNet, GoogleNet, VGG, Resnet, DenseNet, MobileNetV2, NasNet	F1 <=95%	Nvidia TitanX GPU	[122]
Plankton Classification	NA	Plankton image classification	WHOI	AlexNet, GoogLeNet, VGG16, ResNet, PyramidNet	86.3% of accuracy	Nvidia Titan X Pascal GPU	[123]
Label-Free Identification of White Blood Cells	ImageStream (Amnis)	Identification of white blood Cells	Collected	AdaBoost, Gradient Boosting (GB), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM)	99% of accuracy	NA	[62]
Label-Free Leukemia Monitoring	ImageStream (Amnis)	Label-free Leukemia monitoring	Collected	linear SVM	98.2% accuracy	NA	[119]

2.5 Summary

In this Chapter, 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, etc. Moreover, we depicted the fundamentals of visual understanding and deep learning, which is essential to understand machine learning in imaging flow cytometry. Furthermore, we discussed the successful examples using deep learning in this field by summarizing the challenges and limitations encountered when applying deep learning in imaging flow cytometry.

IIFC has shown a board usage in environment monitoring, clinical diagnostics, and other biosensing applications. IIFC needs to improve the quality of imaging to reveal more distinctive features in bioparticles images while maintaining high throughput. Various imaging modalities were proposed to satisfy this goal such as optofluidic time-stretch microscopy. It enables sub-micrometer resolution with clear visualization of internal cell structures without compromising the throughput of cell imaging [127]. In order to apply deep learning for intelligence imaging flow cytometry, immersive datasets are required to train the deep learning model to obtain a high accuracy classifier. Unfortunately, the development of large-scale dataset for imaging flow cytometry is quite challenging. The labelling requires intensive input from the experts to improve the productivity and quality of the labelling. 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, faster and more efficient deep learning models such as SENet [83], and MobileNetV2 [82], etc., have attracted great interest of research community to achieve comparable

classification accuracy on lower-cost hardware. Furthermore, as IIFC will be mass deployed to board area, the cost of the whole system will be considered in the future. Efficient deep learning models such as deep learning model for mobile processor or low-bits deep learning model for intelligent image flow cytometry is worthy of interest.

Chapter 3 Deep Learning-Enabled High-Speed Bioparticle Detection

Imaging flow cytometry is a popular technology for bioparticle images analysis because of its capability of capturing thousands of images per second. Nevertheless, the vast number of images generated by imaging flow cytometry imposes great challenges for data analysis especially when the species have similar morphologies. In this Chapter, a deep learning-enabled high-throughput system for predicting *Cryptosporidium* and *Giardia* in drinking water using imaging flow cytometry is reported. An efficient artificial neural network called MCellNet is demonstrated, which achieves a high classification accuracy $> 99.6\%$, a sensitivity of 97.37% and a specificity of 99.95% . The high-speed analysis reaches 346 frames per second, outperforming the state-of-the-art deep learning algorithm MobileNetV2 in speed (251 frames per second) with a comparable classification accuracy. It has great potential in clinical diagnostics, environmental monitoring and other biosensing applications.

3.1 Introduction

Imaging-based flow cytometry is a popular tool for bioparticle analysis and diseases diagnosis in recent years [128, 129]. Newly emerging imaging flow cytometers (IFC) are capable of capturing thousands to millions of images per second using various imaging modalities [33, 96, 97, 130]. However, the vast number of images with rich information of bioparticle morphologies which imposes a great challenge for data analysis such as classifying different species with similar morphologies by manual gating [131]. For example, the IDEAS-software [132] in **Figure 3-1** uses length and florescent intensities of florescent images to illustrate the target bioparticles. In the gating processing, First, the collected bioparticles are plotted to 2D space with features such as intensities in florescent channel 2 and side scatter channel 6 as shown in **Figure 3-2**. Then, the target bioparticles (*Cryptosporidium* and *Giardia* [133]) are gated by expert with eclipse *Cryptosporidium* and *Giardia* (CG) area as shown in **Figure 3-2**. The eclipse CG area is further separated to individual *Cryptosporidium* (C) and *Giardia* (G) areas in the **Figure 3-3**. Finally, the histogram of C and G area are shown in the **Figure 3-4**. Conventional imaging flow cytometry captures a vast number of images with rich information of bioparticle morphologies, however, imposes a great challenge for data analysis such as classifying different species with similar morphologies.

Automated bioparticle analysis via computer vision and machine learning has greatly free the scientists from cumbersome work on bioparticle identification [134]. Traditional machine learning uses feature engineering for automated bioparticle detection, such as using the feature finder algorithms of IDEAS-software on commercial IFC (Amnis® ImageStreamX Mk II Imaging Flow Cytometry, Luminex Corporation, Austin, United States [96]) to extract features and using classification

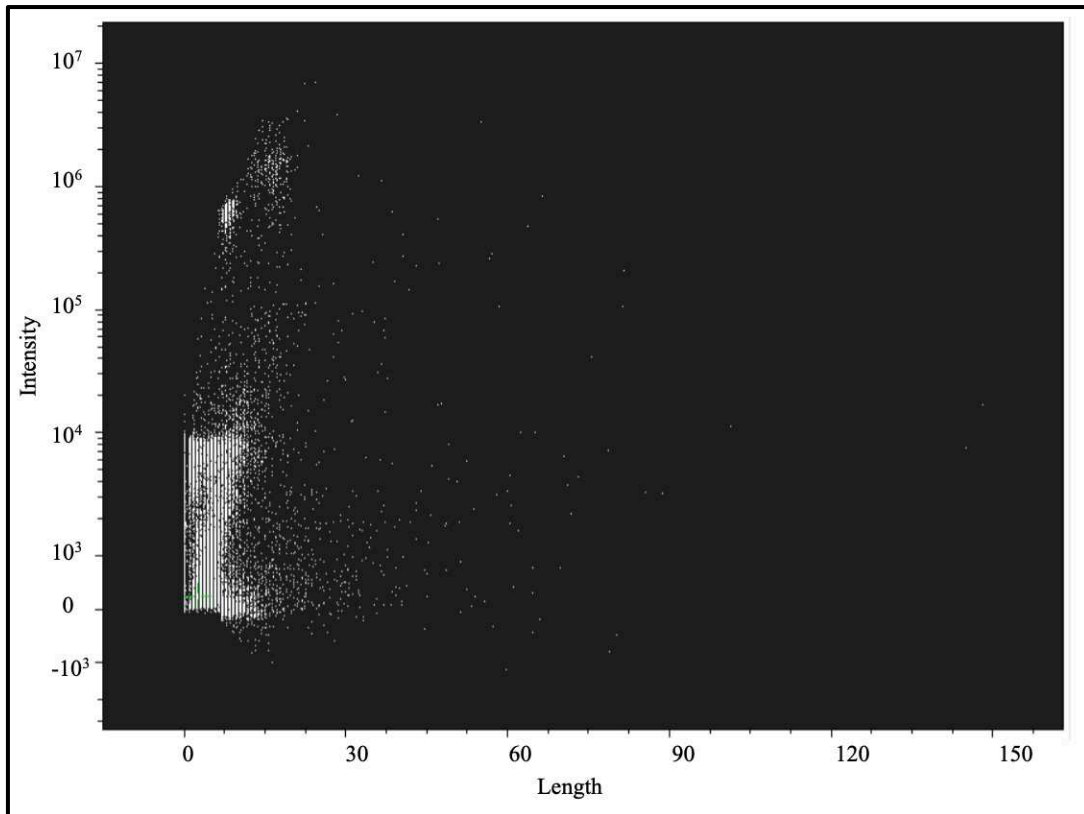


Figure 3-1: Collected bioparticles which plotted by length and florescent intensity.

algorithms (*e.g.*, Principal component analysis, linear discriminant analysis, SVM, random forests and neural networks) to build a bioparticle classification system [135]. Unfortunately, designing features by hand requires rich experience such as profound understanding of the subject and data. It also requires heavy work of debugging. Another difficulty is that machine learning engineers need to design both features and suitable classifier algorithm [136].

Alternatively, deep learning has been widely used to identify bioparticles (*e.g.*, cells) and planktons during image analysis by end-to-end learning [3, 8, 23-29, 47, 137]. For example, Luminex released an AI image analysis software for its imaging flow cytometry [137]. An imaging analysis algorithm based on convolutional neural network was demonstrated on imaging flow cytometry [8]. A deep convolution neural network-enabled image-activated cell sorter was reported with a processing speed of 100 events per second [26], but it required a complex and expensive hybrid

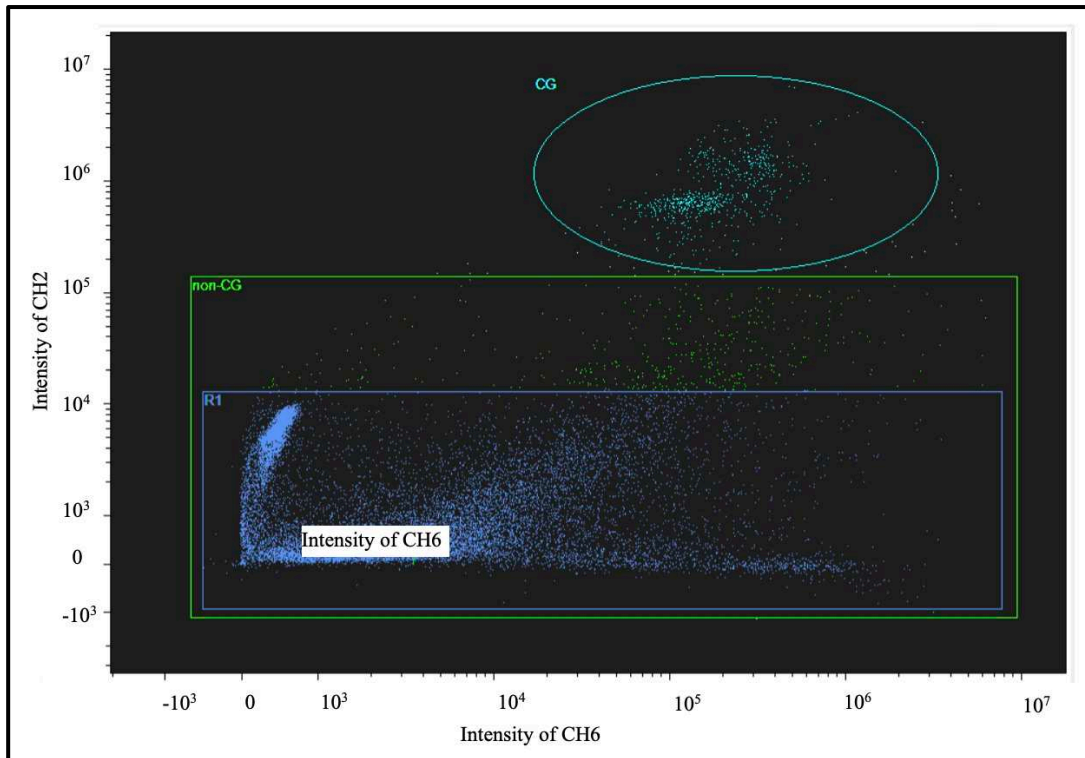


Figure 3-2: The selected CG area (*Cryptosporidium* and *Giardia*) with intensity by expert.

hardware/software data processing system (i.e., a field-programmable gate array (FPGA), three central processing units (CPUs), and a graphics processing unit (GPU)) to ensure a faster computation capability.

With increasingly large image datasets and powerful hardware, several deep learning models such as AlexNet [70], ResNet [81], and DenseNets [40], etc., were reported. They were adopted from the winners of ImageNet [70]. For example, ResNet was the first place of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015. And DenseNet showed a more advanced performance comparing to ResNet. Those deep learning models were also introduced to IFC for detection and classification tasks [4, 5, 40, 48]. They achieved high prediction accuracies of $> 95\%$ but required relatively high-cost system, e.g., high-cost GPUs and CPU comparing to an embedded GPU system. For instance, the Intelligent Image-Activated Cell Sorting 2.0 that achieved a high throughput cell sorting with 2000 cells per seconds but

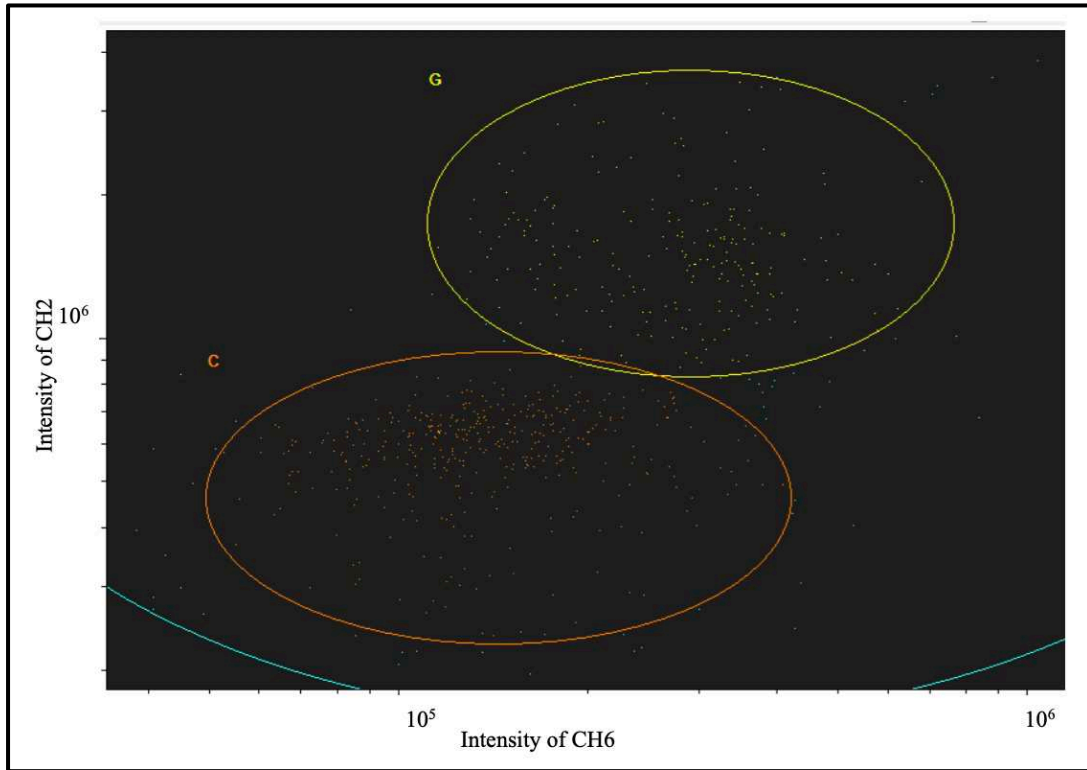


Figure 3-3: The fine selected C (*Cryptosporidium*) area and G (*Giardia*) area.

required an 8-PC server with 8-multicore CPUs and GPUs for image processing using deep learning [4]. Deep learning models have also been used on IFC for water quality assessment such as detection of algae [138-140].

Most existing deep learning algorithms developed for IFC mainly focus on the detection accuracy but overlook the trade-off between speed and hardware requirements. High-end GPU system empowers the training of complex deep neural networks, but it is a major hurdle for mass deployment of these deep learning algorithms to commercial IFC for bioparticle analysis due to its high cost and high-power consumption. Very recently, faster and efficient deep learning models, such as MobileNet [86], SENet [83], and MobileNetV2 [82], have attracted great interest from the research community because they are able to achieve comparable classification accuracies using lower-cost embedded hardware. However, they need extensive optimizations for specific tasks.

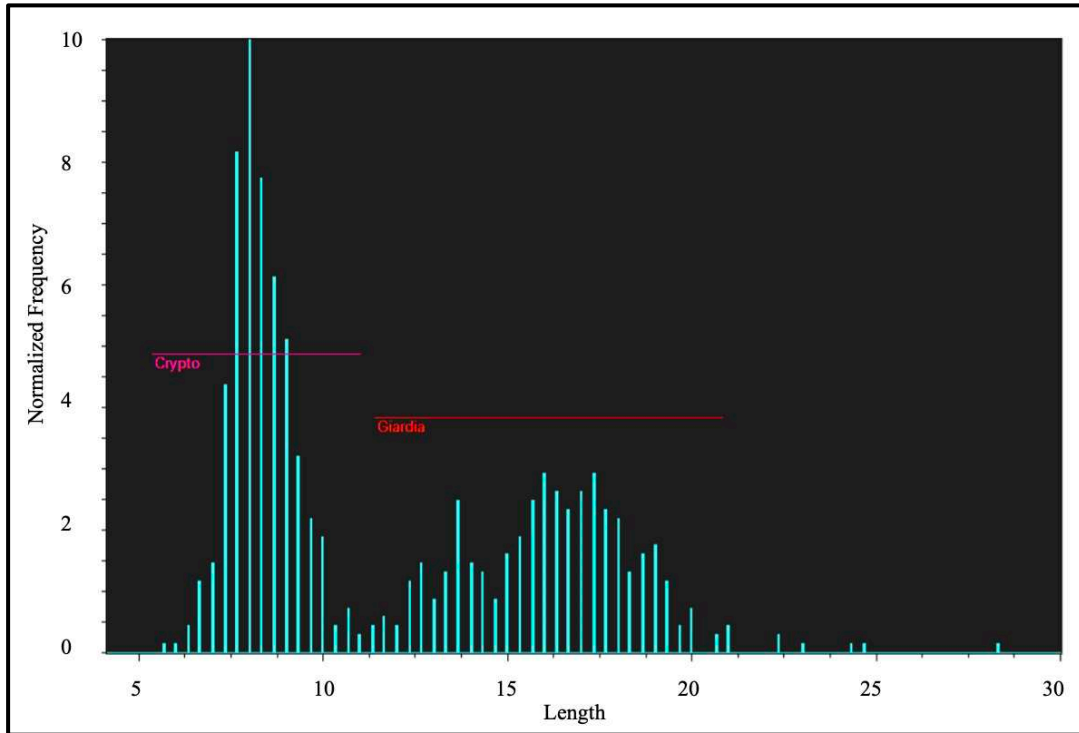


Figure 3-4: The histogram graph of the selected *Cryptosporidium* and *Giardia*.

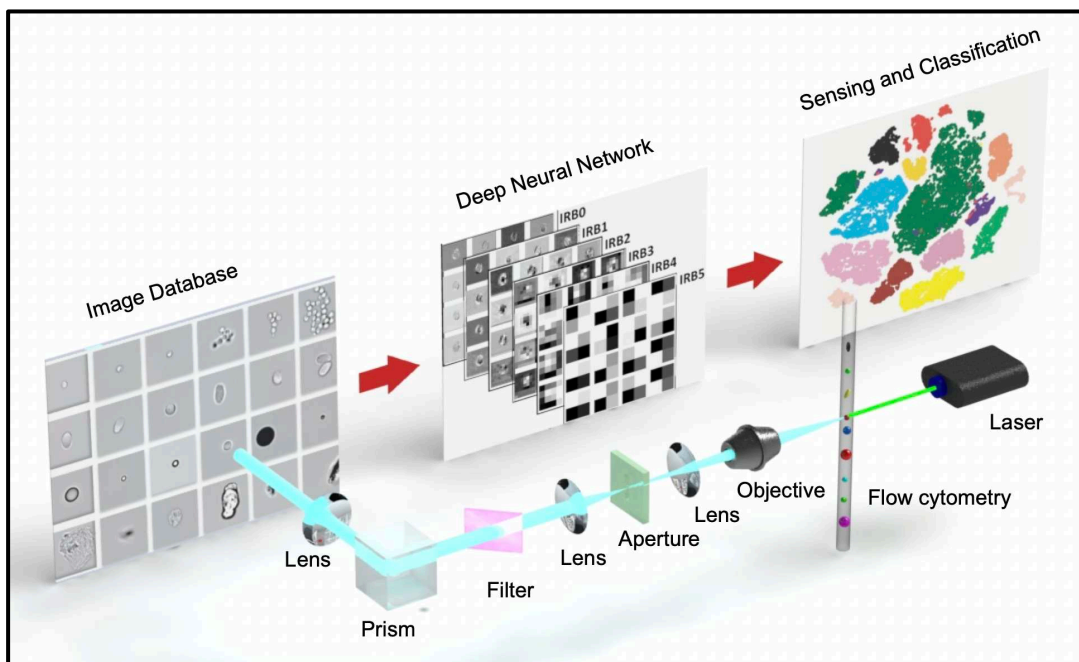


Figure 3-5: Overview of MCellNet, a deep neural network assists the imaging flow cytometry (Amnis® ImageStream®X Mk II) in *Cryptosporidium* and *Giardia* detection. The system consists of laser, flow cytometry, imaging system, image database and deep neural network.

In this Chapter, an efficient neural network called MCellNet for the high-speed detection and classification of *Cryptosporidium* and *Giardia* from microplastics and other pollutants in the water using the imaging flow cytometry is proposed as shown in **Figure 3-5**. Bioparticles were hydrodynamically focused by a sheath flow and flowed through the detection region with Phosphate Buffered Saline solution (PBS) as the sheath medium. Single bioparticles were illuminated with a LED light source [141], and bright-field images were acquired with a Charge-Coupled Device (CCD) camera [142] using a $60\times$ objective. Then, the MCellNet used the building block of a widely adopted efficient deep network, MobileNetV2, and optimize it to increase processing speed and accuracy for imaging flow cytometry.

MCellNet image processing pipeline is a software that can detect *Cryptosporidium*, *Giardia* and microplastics using two methods, i.e., multiclass and binary class classifications, respectively. Multiclass classification is used to classify instances into 13 classes: *Cryptosporidium*, *Giardia*, $1.54\text{-}\mu\text{m}$ beads, $3\text{-}\mu\text{m}$ beads, $4\text{-}\mu\text{m}$ beads, $4.6\text{-}\mu\text{m}$ beads, $5\text{-}\mu\text{m}$ beads, $5.64\text{-}\mu\text{m}$ beads, $8\text{-}\mu\text{m}$ beads, $10\text{-}\mu\text{m}$ beads, $12\text{-}\mu\text{m}$ beads, $15\text{-}\mu\text{m}$ beads, and natural pollutants. And the binary classification is used to classify instances into two classes with the natural pollutant and microplastics in one class (Class 0), and all *Cryptosporidium* and *Giardia* images in another class (Class 1). The pipeline works by reading the raw data of bright-field images extracted by IDEAS software and patching to images with 120×120 pixels, then putting into MCellNet to generate classification output.

This MCellNet only requires an embedded processor with low computational power such as Nvidia Jetson TX2 [44], thus can be used in affordable machines. By combining MCellNet and imaging flow cytometry, the proposed system achieves a detection speed of 346 frames per second, outperforming the state-of-the-art deep

learning algorithm MobileNetV2 (251 frames per second) under a comparable classification accuracy. *Cryptosporidium* and *Giardia* which are common pathogens in drinking water to cause gastrointestinal diseases and have already become a major source of pollution in drinking water and food industries [133, 143-145], are used as the case study to validate our system. Our testing results show that MCellNet significantly enhances both the accuracy and processing speed, allowing it to be incorporated in drinking water quality inspection. The system also empowers potential high-throughput bioparticle analysis applications in environmental monitoring [50], clinical diagnostics [49], and other biomedical applications.

3.2 Model for High-speed Bioparticle Detection

3.2.1 Overall Structure of MCellNet

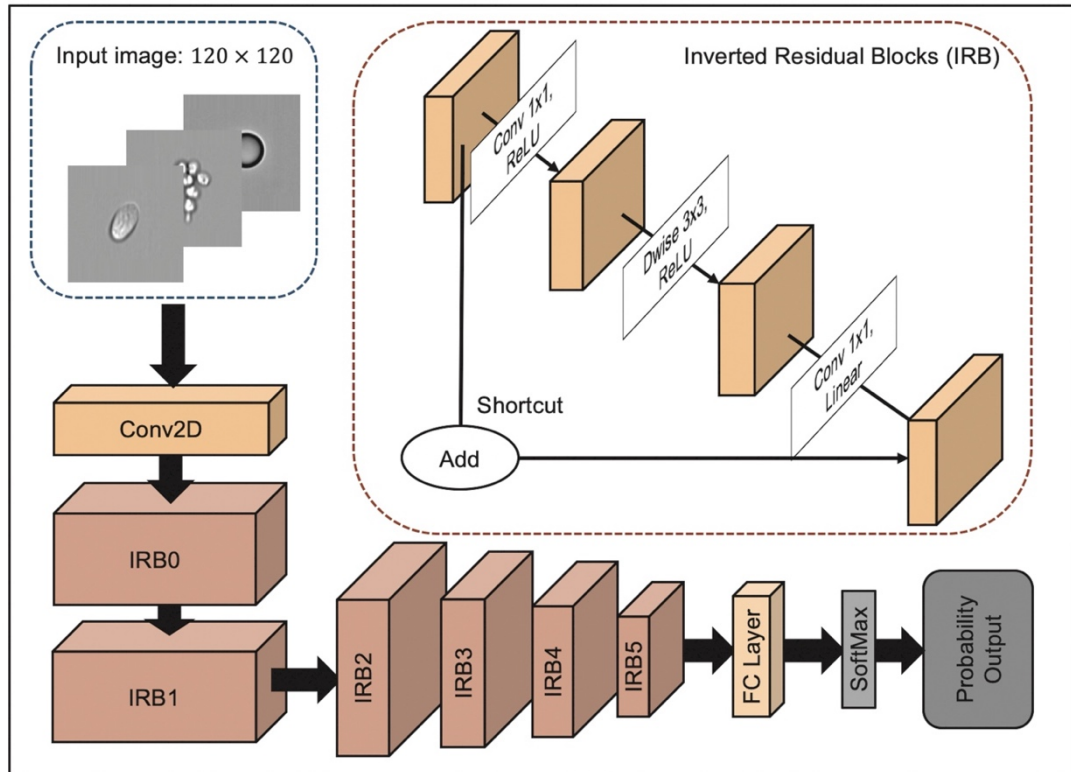


Figure 3-6: MCellNet architecture, which contains one convolutional layer, 6 inverted residual blocks, one flattened layer, and one fully connected layer. Each IRB forms a shortcut between the bottlenecks to perform identity mapping.

MCellNet is an artificial intelligence neural network to distinguish the *Cryptosporidium*, *Giardia*, microplastics and water pollutants from bright-field images. MCellNet is developed by cascading six computation-efficient neural network blocks - Inverted Residual Blocks (IRBs) to achieve a high speed and accuracy. IRBs were introduced by MobileNetV2 with lower computational complexity and maintain high efficiency representational power.

The overall structure of MCellNet is illustrated in **Figure 3-6** and **Table 3-1**. The input of MCellNet is bright-field images of 120×120 pixels, and the output is

a probability vector \mathbf{y} that indicated the likelihood of the presence of each class.

Table 3-1: Network parameters

Layer/Block	Type	Output dimension	Params
Conv2d	Convolution	$60 \times 60 \times 24$	744
IRB0	Inverted Residual Block	$60 \times 60 \times 24$	760
IRB1	Inverted Residual Block	$30 \times 30 \times 24$	5568
IRB2	Inverted Residual Block	$15 \times 15 \times 24$	9456
IRB3	Inverted Residual Block	$8 \times 8 \times 24$	9456
IRB4	Inverted Residual Block	$4 \times 4 \times 24$	9456
IRB5	Inverted Residual Block	$2 \times 2 \times 16$	8272
Dense1	Fully Connected	13	845

The working principle of MCellNet is as follows. First, MCellNet consists of one convolutional layer with a filter size of 3×3 pixel. Then six IRBs (IRB0-5) is attached to the first convolution layer. Later, the output of the last IRB (IRB5) is flattened. Finally, a Fully-Connected (FC-13/FC-2) layer [136] with 13 output units for multiple classification or 2 output units for binary classification are attached to IRB5's output to generate the class score for bioparticles detection.

The first convolutional 2D layer (Block Conv2D in **Figure 3-6**) [146] takes an $h \times w \times n$ input feature map \mathbf{X}^i , where h is the spatial height, w is the spatial width and the n is the output channels of the feature map, and it is $120 \times 120 \times 1$ in this case. The input is transformed into a $64 \times 64 \times 24$ output feature maps \mathbf{X}^o . The input feature map \mathbf{X}^i is convoluted with a number of feature detectors, each of which is a three-dimensional filter \mathbf{F} in the present layer, and a bias \mathbf{b} . An activation function $\delta(x)$, ReLU in our case, is attached to this convolution operator. The whole layer can be expressed as [86]

$$\mathbf{X}_{x,y,z}^o = \delta\left(\sum_{i,j,k} \mathbf{F}_{i,j,k,z} \cdot \mathbf{X}_{x+i-1,y+j-1,k}^i + \mathbf{b}_z\right) \quad (3-1)$$

where $z = 1, 2, \dots, m$ and $k = 1, 2, \dots, n$.

The six IRBs (IRB0-5) attach to the first convolution layer is shown as IRB0-5 in **Figure 3-6**. Each IRB is based on the inverted residual containing a narrow-wide-narrow structure. Inside the IRB (sub graph inside the **Figure 3-6**), there are a 1×1 expansion convolutional layer, a depthwise convolution layer and a 1×1 projection layer. The depthwise convolution and 1×1 projection layer are referred to as the depthwise separable convolution. The depthwise separable convolution was adopted by Xception [88] which consisted of the depthwise convolution followed by a pointwise convolution. The depthwise convolution can be represented as [86]

$$\hat{X}_{x,y,z}^k = \delta(\sum_{i,j} \hat{F}_{i,j,z}^k \cdot X_{x+i-1,y+j-1,z}^{k-1} + b_z^k) \quad (3-2)$$

where $\hat{F}_{i,j,z}^k$ is the depthwise filter in which the z_{th} channel in $\hat{F}_{i,j,z}^k$ only calculates with the z_{th} channel of $X_{x+i-1,y+j-1,z}^{k-1}$ and produces the feature $\hat{X}_{x,y,z}^k$ in the z_{th} channel.

Compared to the traditional convolution, the computational reduction of the depthwise separable convolution is $1/N + 1/D_k^2$, where N is the number of output channels, and D_k is the kernel size. Furthermore, the resource efficiency of IRB also increases the resource efficiency with its unique architecture. In addition, the skip connection structure is introduced to IRB, which gives the network an opportunity to access features in earlier stages and lead to a deeper neural network with a high efficiency.

The classification layer is a Fully-Connected layer (FC Layer in **Figure 3-6**). It takes the last output of IRB5 as the input and applies matrix multiplications to the weight matrix F to produce the number of output classes (2 or 13). This operation generates class scores to distinguish the targets.

The output equation of this layer can be expressed as

$$\mathbf{X}^o = \mathbf{F}\mathbf{X}^i + \mathbf{b} \quad (3-3)$$

Softmax regression[75] of the classification layer outputs is employed to train the network. The output of Softmax regression \mathbf{y} can be written as

$$y_j = \frac{\exp\{x_j^k\}}{\sum_{i=1}^n \exp\{x_i^k\}}, j = 1, 2, \dots, n \quad (3-4)$$

3.2.2 Training of the Model

To train MCellNet and evaluate its performance in imaging-based high-speed *Cryptosporidium*, *Giardia* and microplastics detection, the image dataset was randomly split with into a training dataset (38,469 images), a validation dataset (9,618 images) and a test dataset (32,059 images) that contained 48%, 12% and 40% of the total number of images by using scikit-learn library [147], respectively. In the task of binary classification, the images of microplastics and natural pollutants were combined into one single class, and the *Cryptosporidium* and *Giardia* images were merged into another. The training, validation and test dataset were used to tune the parameters of MCellNet during the training phase, tune the hyper-parameters, and assess the performance of MCellNet, respectively.

Deep neural networks are implemented with TensorFlow [148] and are trained over an Ubuntu GPU server [149] with four GPU cards (Nvidia GeForce RTX 2080 Ti) and one Intel CPU (Xeon E5-2650). The weight matrices of the deep neural networks (\mathbf{F} and \mathbf{b}) are initialized with the Glorot uniform initializer [150], and the networks (\mathbf{F} and \mathbf{b}) are trained in an end-to-end fashion using the Adam stochastic optimizing algorithm [151]. The parameters for Adam are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate decay is used for training. A data augmentation mechanism [152] is used in the training process to enhance the training dataset size. The images are augmented by random position transformation, horizontal and vertical flipping, and

image rotation and zooming during training. In addition, a class weights [153], parameter space search over the entire training phase with a learning rate and learning rate decay are also performed.

3.2.3 Optimization of the Model

Deep learning promises accurate and rapid imaging-based analysis of *Cryptosporidium*, *Giardia* and microplastics of different sizes. Nevertheless, existing works in this field rely on computer servers with powerful accelerator and fat neural network architecture [36] that outstrips the capability of many mobile and embedded processors for field biomedical applications. It is therefore necessary to develop efficient mobile neural networks with reduced hardware footprint and computing power requirements. Efficient neural network architecture design is a tradeoff between the accuracy and speed by carefully tuning the depth (d), width (w) and resolution (r) of the neural network [154]. The target of optimization is to retain a high accuracy while reducing the floating-point operations (FLOPS) which is a key indicator of neural network efficiency. Previous study [154] suggests that FLOPS of a neural network is proportional to d , w^2 and r^2 . A practical approach of optimizing a neural network starts with a handcrafted base neural network block and evolves it through an architecture search. According to this principle, we have designed an MCellNet based on IRB with a parameter search for trade-off between accuracy and efficiency.

To find the optimal IRB numbers for the MCellNet, an IRB search is also performed. Because our dataset is less complex than ImageNet, a high-level block search with the goal to reduce w and d are conducted on the baseline MobileNetV2 architecture for neural network optimization. The input resolution of the neural network is kept constant. The width of the neural network is reduced to search for a

suitable network scale for our tasks, and the depth of the neural network is reduced to further decrease the FLOPS. Several models with different number of IRB blocks ranging from 17 to 3 are built and tested (**Table 3-2**). The optimized configuration has 6 IRBs that shows a comparable accuracy score to the original MobileNetV2 (17 IRBs) with lesser computation power and with faster processing speed on this dataset.

Table 3-2: IRBs search for base network

Methods	Number of IRBs	Images per Second	Accuracy
MobileNetV2	17	258.4	99.4697%
Model1	10	301.5	99.6225%
Model2	8	314.7	99.6849%
MNetSearch	6	319.7	99.5134%
Model4	3	274	99.4635%

In order to further regularize the neural network, the stride parameters are adjusted on every IRB in addition to the network search. With this modification, the depth and width of MCellNet is shrunken from 120×120 to 2×2 . Furthermore, a Fully-Connected (FC) layer is added to combine the structural features from the underlying features map. The above restructuring leads to MCellNetA with a 99.64% accuracy at a processing speed of 346 images per second, comparing with the best optimized MobileNetV2 in **Table 3-2** of the accuracy of 99.51% at the speed of 319 images per second on Nvidia Jetson TX2.

3.3 Experimental Results and Discussions

3.3.1 Image Acquisition of Bioparticles using Imaging Flow Cytometry

Cytometry

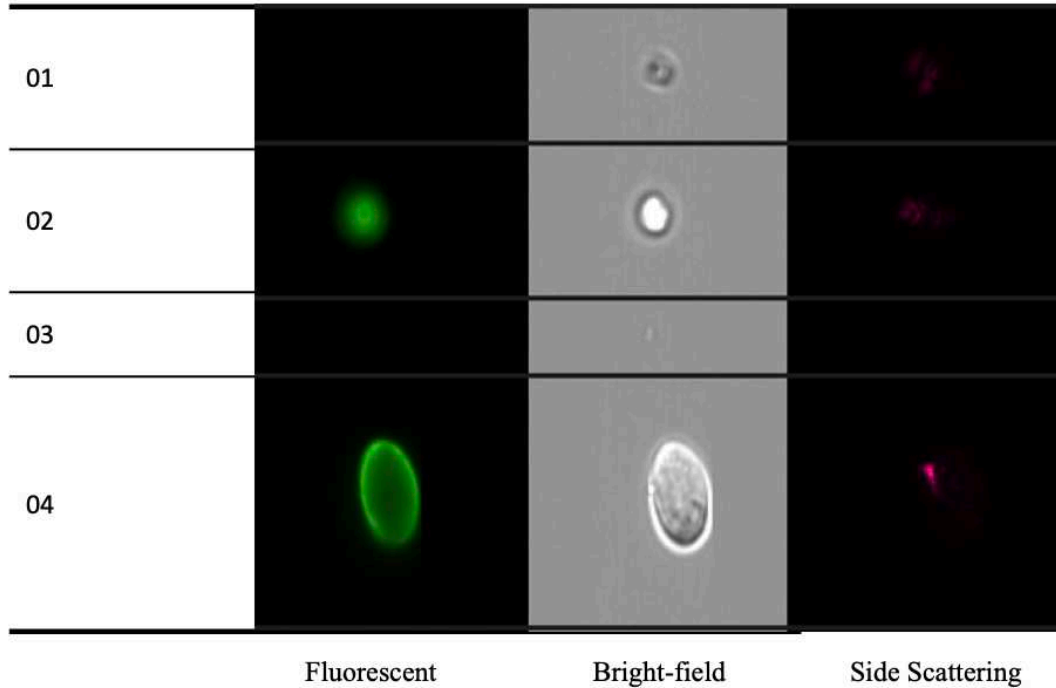


Figure 3-7: Raw image sequence of bioparticles. Each line represents one bioparticle's images. From left to right is fluorescent image, bright-field image, and side scattering image.

Cryptosporidium, *Giardia*, microplastics and pollutants such as dirt and cell debris with size from 3 to 14 μm that naturally exist in drinking water were included in the study. The naturally existing pollutants were obtained by concentrating 10 liters of drinking water using a water filtration system. Formalin-treated *Cryptosporidium*, *Giardia* oocyst (Waterbornetm Inc) and synthetic microplastic beads (from Thermo Fisher Scientific, Duke Scientific and Polysciences Inc.) of different sizes (1.54 μm , 3 μm , 4 μm , 4.6 μm , 5 μm , 5.64 μm , 8 μm , 10 μm , 12 μm , and 15 μm) samples were spiked separately into 200 μL water. Images of microparticles in spiked water samples were observed by imaging flow cytometry (Amnis[®] ImageStream^X Mk II

Imaging Flow Cytometry, Luminex Corporation, Austin, United States) [96]. Bioparticles such as *Cryptosporidium* and *Giardia oocysts*, microplastics and naturally existing particulate pollutants were hydrodynamically focused by a sheath flow and flowed through the detection region. The bright-field image of bioparticle was acquired by a CCD camera [142] using a $60\times$ objective with illuminated with a LED light source.

The raw image sequence files (.RIF) of different samples were captured as shown in **Figure 3-7**. Each line represents one type of bioparticle's imaging. From left to right are fluorescent imaging, bright-field imaging, and side scattering imaging. The raw bright-field images are extracted from the image sequence files by IDEAS software (accompanying with the ImageStream) and each raw image is patched to 120×120 pixels as in **Figure 3-8**. The first row is *Cryptosporidium*, the second row is *Giardia*, the third row is microbeads, and the last row is natural pollutants.

From millions of raw images acquired, 80,146 images were selected by experts for building the image database. The selected raw images were manually labeled and put into a database that consisted of 13 classes: *Cryptosporidium* (2,082 images, **Figure 3-9a**), *Giardia* (3,569 images, **Figure 3-9b**), $1.54\text{-}\mu\text{m}$ beads (3,466 images, **Figure 3-9c**), $3\text{-}\mu\text{m}$ beads (3,457 images, **Figure 3-9d**), $4\text{-}\mu\text{m}$ beads (5,783 images, **Figure 3-9e**), $4.6\text{-}\mu\text{m}$ beads (2,188 images, **Figure 3-9f**), $5\text{-}\mu\text{m}$ beads (9,637 images, **Figure 3-9g**), $5.64\text{-}\mu\text{m}$ beads (3,285 images, **Figure 3-9h**), $8\text{-}\mu\text{m}$ beads (3,066 images, **Figure 3-9i**), $10\text{-}\mu\text{m}$ (8,270 images, **Figure 3-9j**), $12\text{-}\mu\text{m}$ (4,704 images, **Figure 3-9k**), $15\text{-}\mu\text{m}$ beads (2,813 images, **Figure 3-9l**), and natural pollutants of various shapes and sizes (27,826 images, **Figure 3-9m**). All figures share the same scale bar in (**Figure 3-9a**).

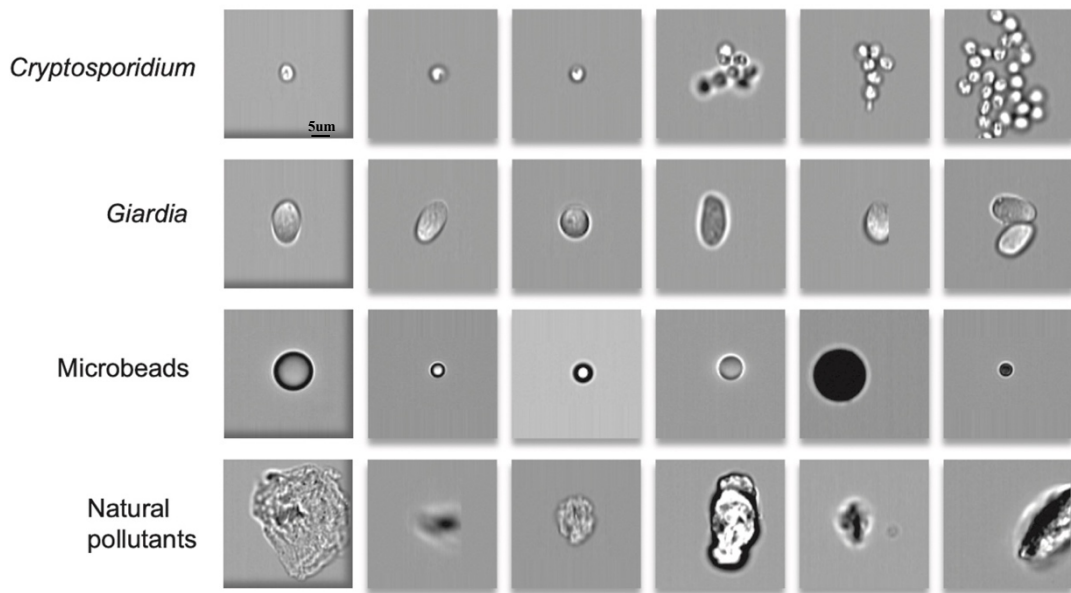


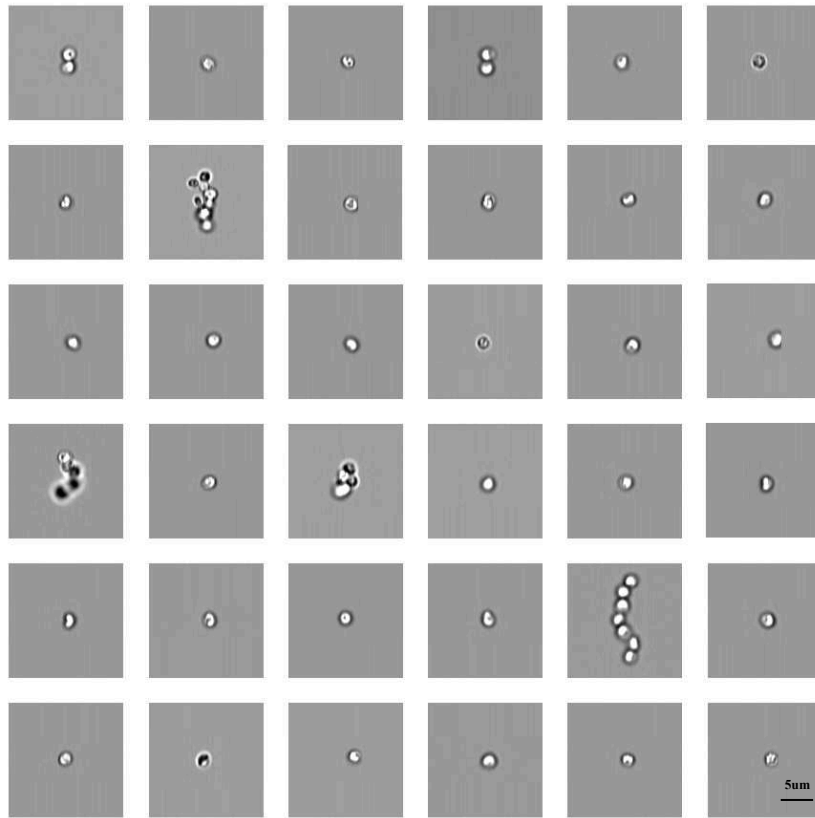
Figure 3-8: Bioparticle image dataset. Each row represents one type of bioparticles. From the top to bottom are *Cryptosporidium*, *Giardia*, microbeads (12 μm , 4 μm , 5.64 μm , 8 μm , 15 μm , and 3 μm .), and natural pollutants. All sub figures share the same scale bar.

The bright-field images of *Cryptosporidium*, *Giardia* and other bioparticles has complex patterns such as distinct size, internal structure, and aggregation. The orientation and relative position of them in the image further complicated the matter (**Figure 3-8**). For example, *Cryptosporidium* and *Giardia* appear in the form of both single and aggregates in the captured images. *Cryptosporidium* appears spherical with a diameter of 3-6 μm , whereas *Giardia* appears oblong with a long axis of 8-14 μm . *Giardia* may appear in various shapes depending on its orientation. Microplastic beads appear sphere with different sizes. The naturally existing pollutants appear in various sizes and shapes. Some images appear a bit blur because they were slightly out of focus even though an autofocus mechanism is applied. For example, the target bioparticle sometime is blur or out of focus as in **Figure 3-10**. The images are separated by a focus metric – gradient RMS score. The left part is the out of focus images and the right part is the focused images. In certain cases, bioparticles at the

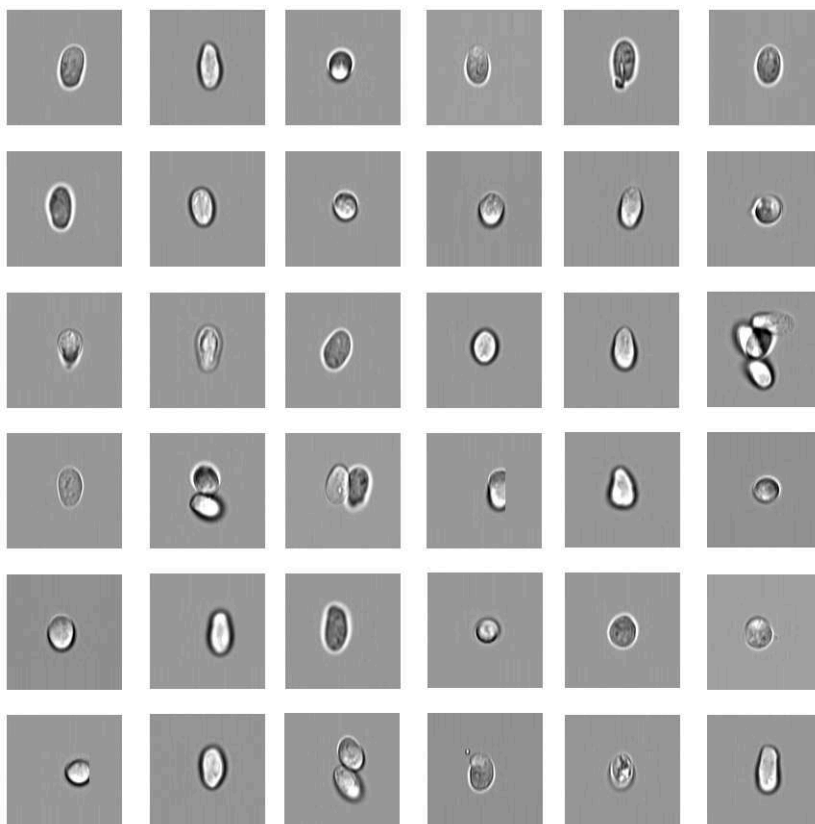
edge of the field of view are partially occluded. Moreover, the fluctuation of illumination conditions and vibration will cause variability to the images, which further complicate the detection task.

3.3.2 Evaluation Metrics

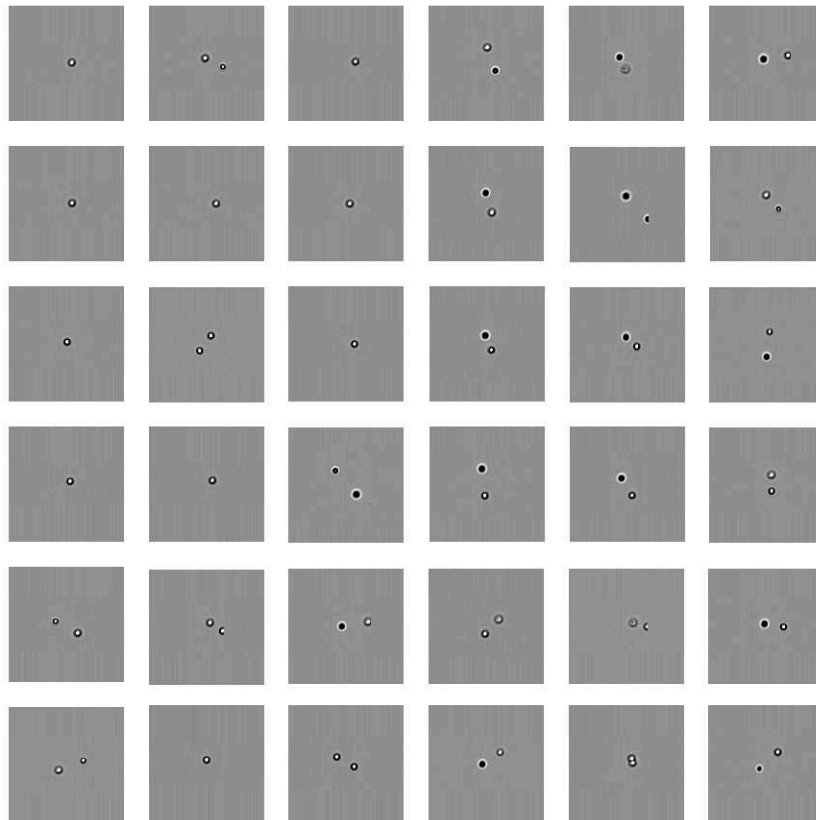
MCellNet is capable of rapid and accurate imaging-based detection of *Cryptosporidium* and *Giardia* over microplastics and natural pollutants under both multiclass classification and binary classification. Multiclass classification is used to classify images into one of three or more classes so we can know detail information on *Cryptosporidium*, *Giardia*, microplastic beads, or other contaminated pollutants. Binary classification is used to evaluate the performance on two classes of classification condition. Binary classification is often used when we need to distinguish high-risk bio-contaminants, *Cryptosporidium* and *Giardia*, from low-risk microplastics and other naturally existing particle pollutants. We compare MCellNet with the state-of-the-art deep neural network MobileNetV2 in both tasks. For processing speed, all the models are evaluated on Nvidia Jetson TX2 (Jetson TX2 Developer Kit, Nvidia Corporation, California, United States).



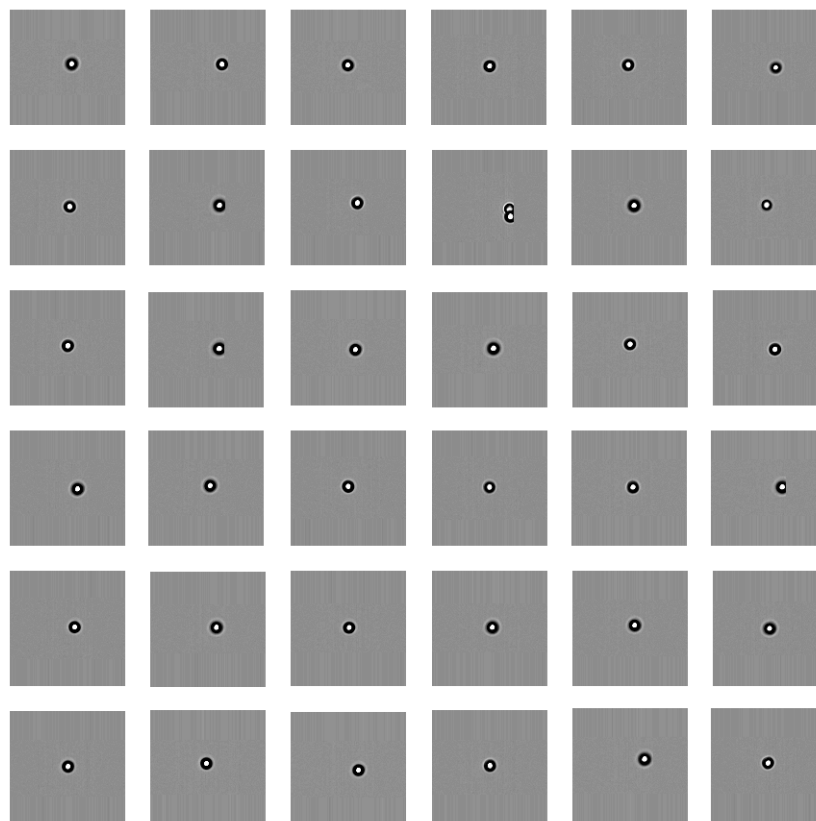
(a)



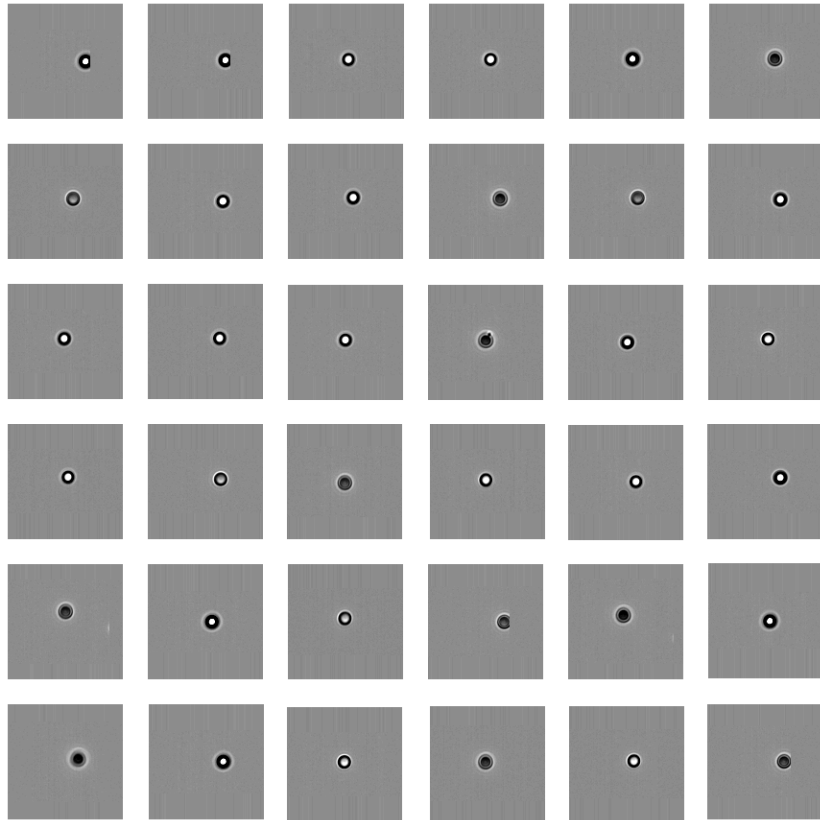
(b)



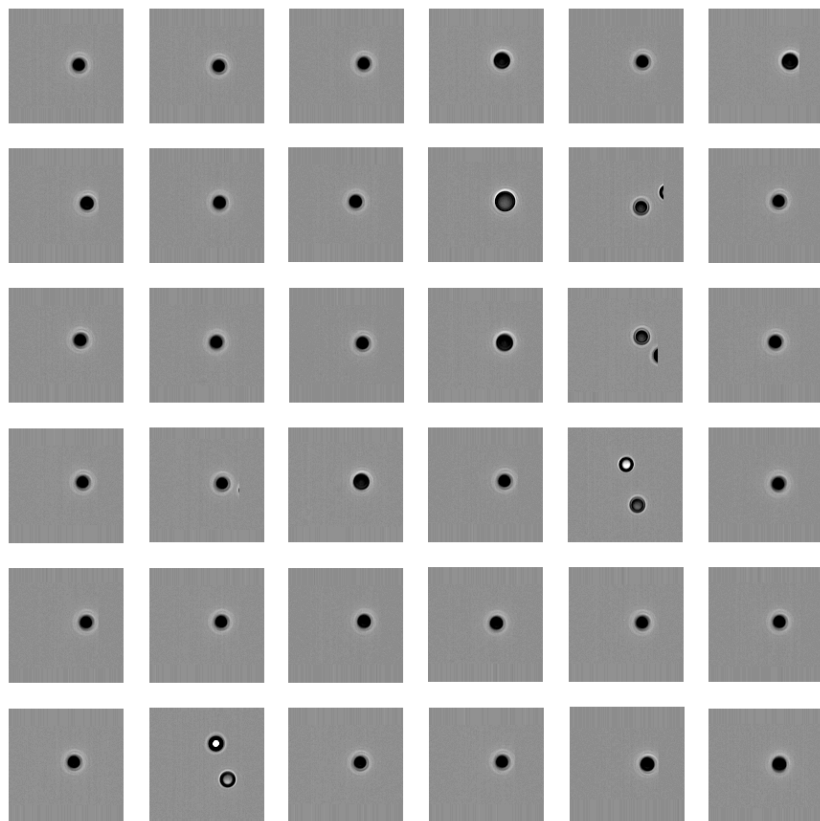
(c)



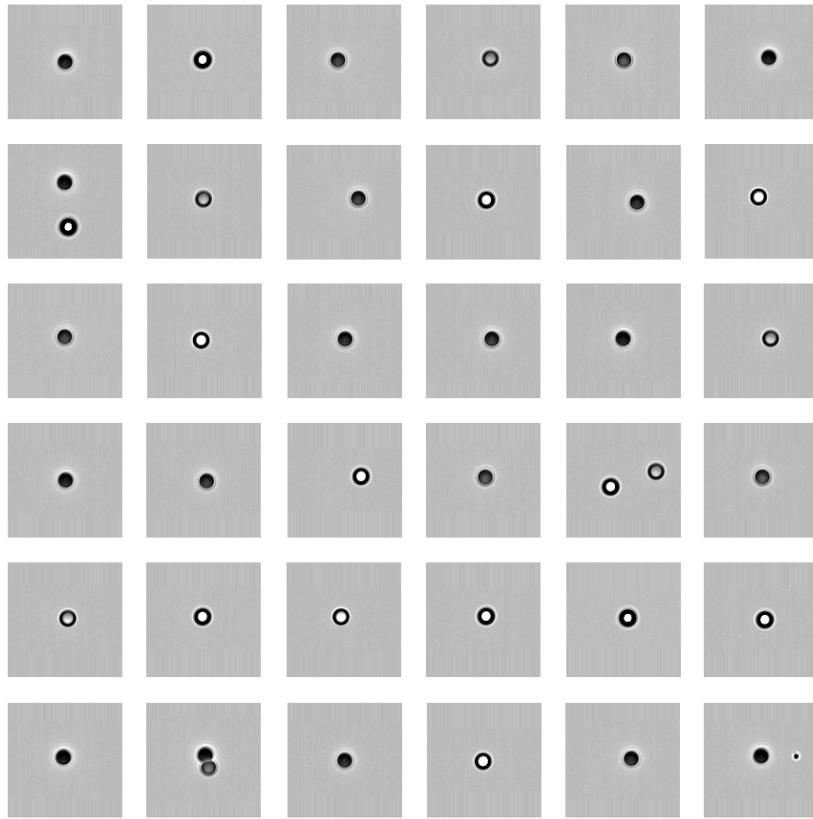
(d)



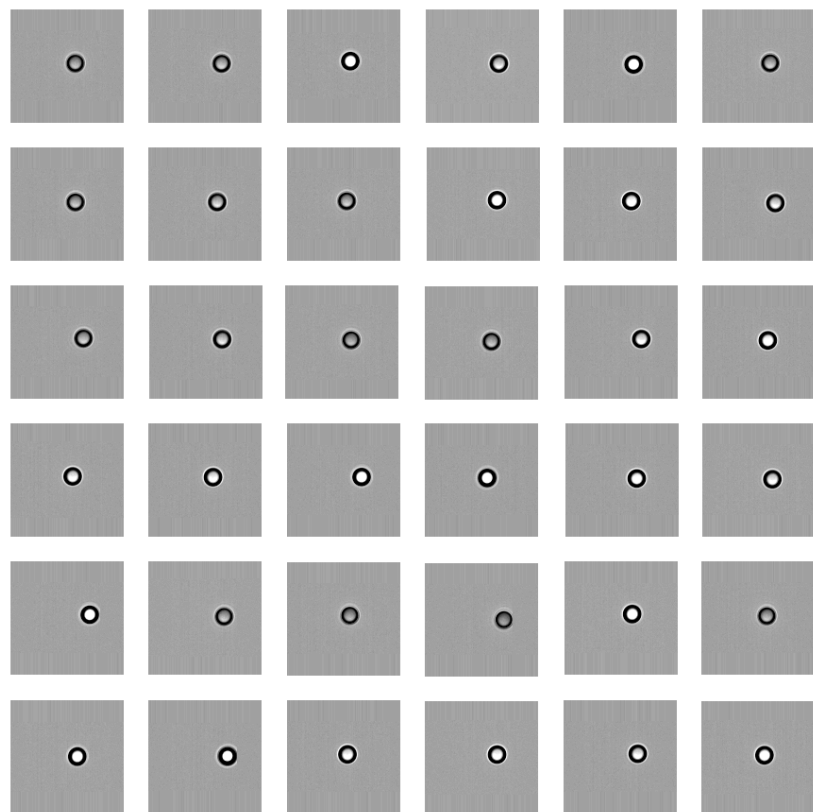
(e)



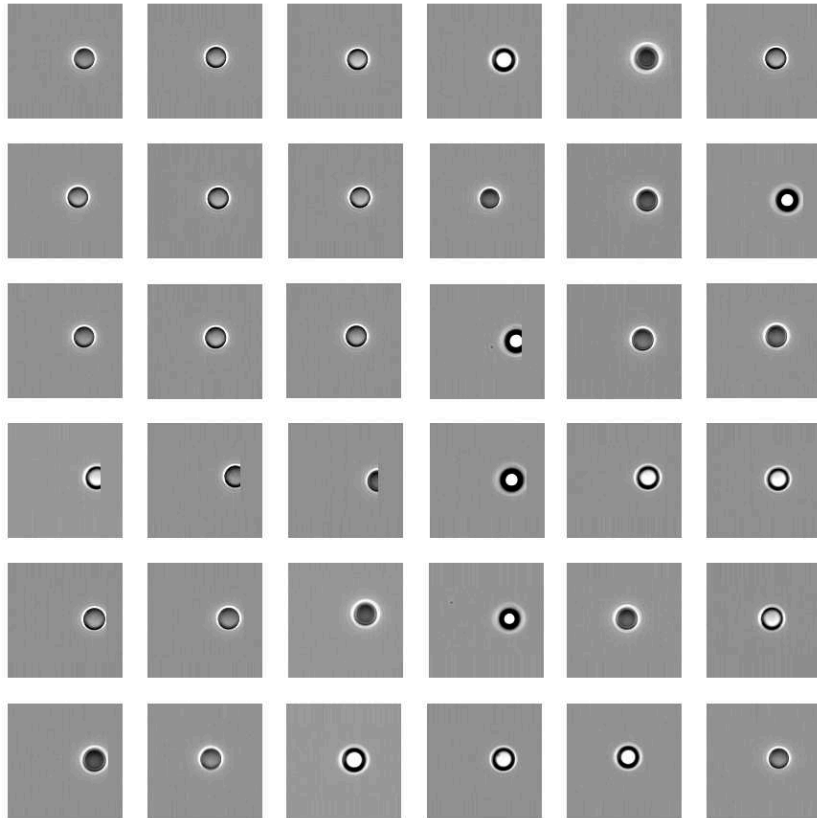
(f)



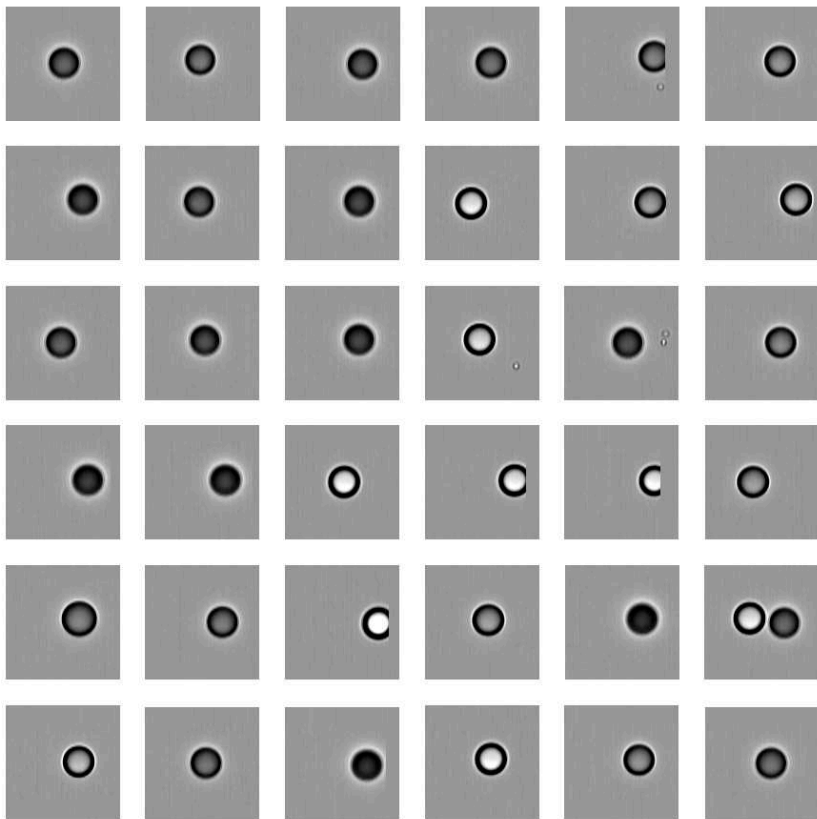
(g)



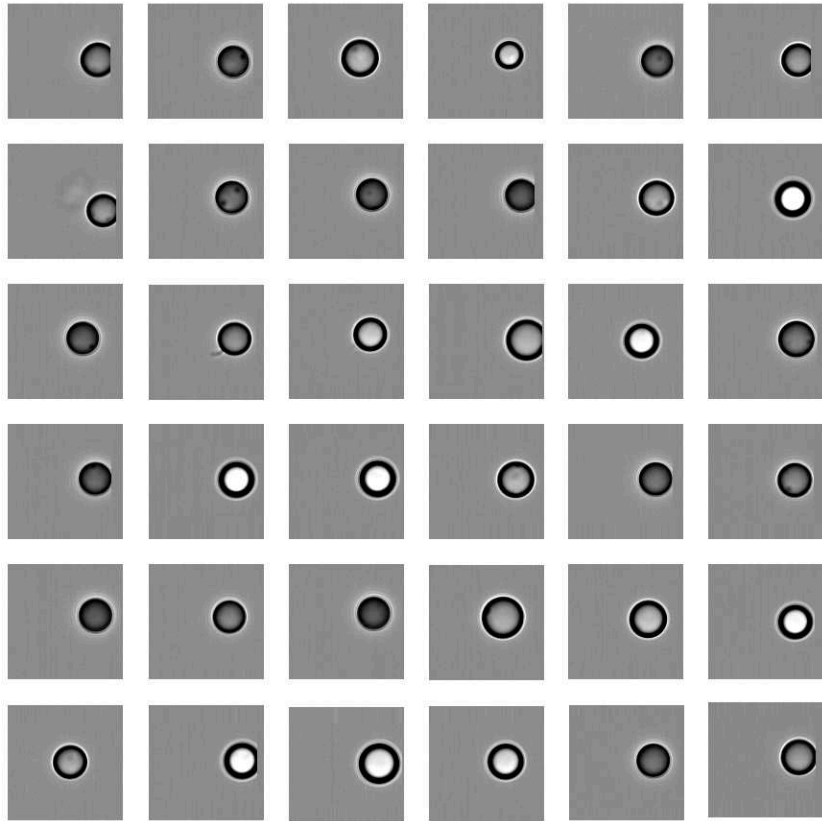
(h)



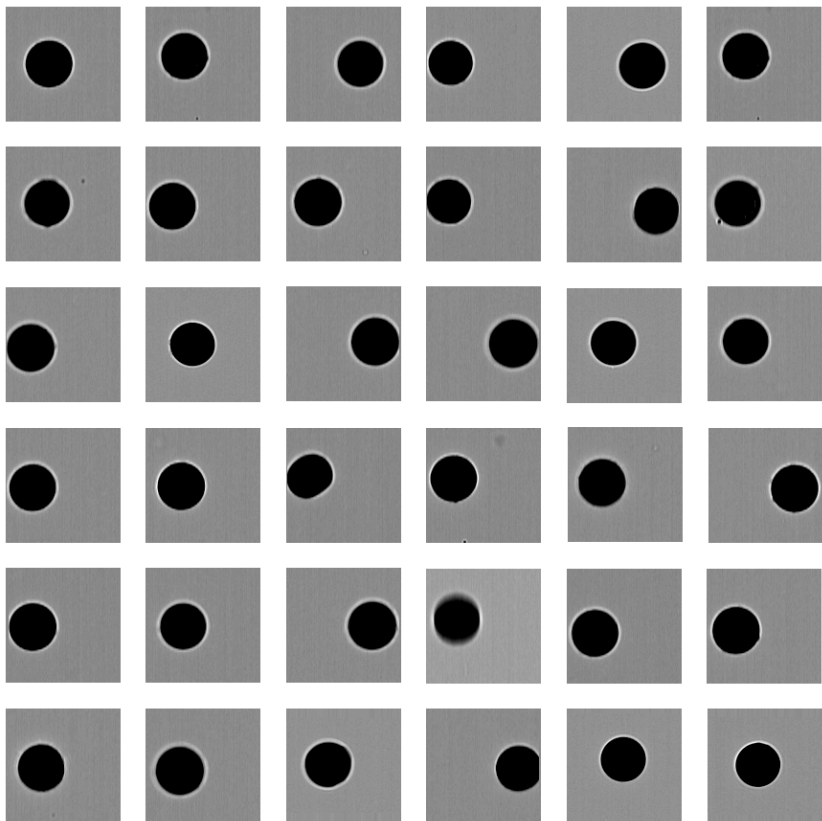
(i)



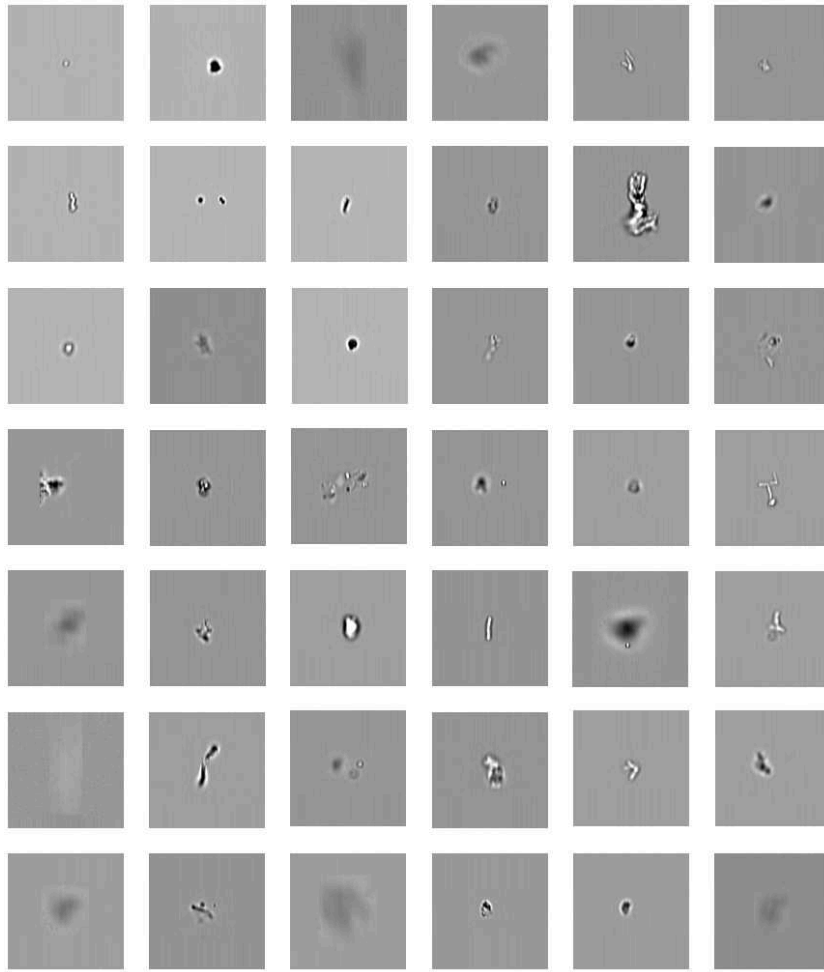
(j)



(k)



(l)



(m)

Figure 3-9: Single bioparticle image dataset. Each block represents one type of bioparticle. From the top to bottom are (a) *Cryptosporidium*, (b) *Giardia*, microbeads with size of (c) $1.54\mu m$, (d) $3\mu m$, (e) $4\mu m$, (f) $4.6\mu m$, (g) $5\mu m$, (h) $5.64\mu m$, (i) $8\mu m$, (j) $10\mu m$, (k) $12\mu m$, and (l) $15\mu m$, and (m) natural pollutants. All figures share the same scale bar in (a).

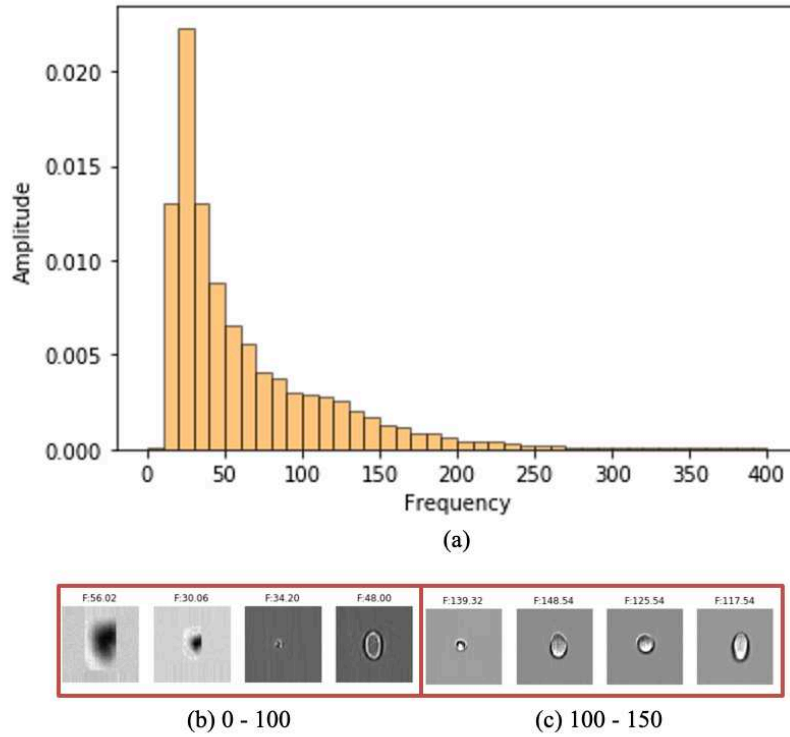


Figure 3-10: Data Distribution on Gradient RMS. (a) The normalized histogram of Gradient RMS. (b) The example images with of lesser RMS (blur) and high RMS (clear).

In order to evaluate the performance of MCellNet, the accuracy, precision, sensitivity or recall, F1-scores, specificity, and images per second are adopted for both multiclass classification and binary classification and listed in the **Table 3-3**.

Table 3-3: Evaluation metrics

Accuracy	$\frac{TP + TN}{P + N}$	(3-5a)
Precision	$\frac{TP}{TP + FP}$	(3-5b)
Sensitivity or recall	$\frac{TP}{P}$	(3-5c)
F1-scores	$2 \times \frac{\textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$	(3-5d)
Specificity	$\frac{TN}{N}$	(3-5e)
False Negative Rate	$\frac{FN}{P}$	(3-5f)
False Positive Rate	$\frac{FP}{N}$	(3-5g)

where TP is true positive, FP is false positive, TN is true negative, FN is false negative, P is condition positive and N is condition negative. Particularly, for multiclass classification, confusion matrices, also known as error matrix, are used for evaluating the performance. Each row of the confusion matrix presents the percentage of predicted class, and each column presents the actual class.

Especially, for the binary classification, the False Negative Rate-False Positive Rate (FNR-FPR) curves are used to show the performance and compare the performance of different approaches at Equal Error Rate (EER), FNR at 0.1% and FPR at 0.1%. EER means the FNR and FPR are equal. Two version of MCellNet

binary classification models (MCellNetA and MCellNetB) are built for binary classification. MCellNetA is a deep neural network evolved directly from MNetSearch. When training MCellNetA, all images of microplastics are merged with the natural pollutant images into one class (Class 0), and all *Cryptosporidium* and *Giardia* images are merged into another class (Class 1). MCellNetB is generated by converting the neural network trained as a multiclass classifier (MCellNetM) into a binary classifier. Instead of merging the raw data into two classes, the output is merged into two classes.

Table 3-4: Average accuracy, precision, recall and F1-score on test dataset

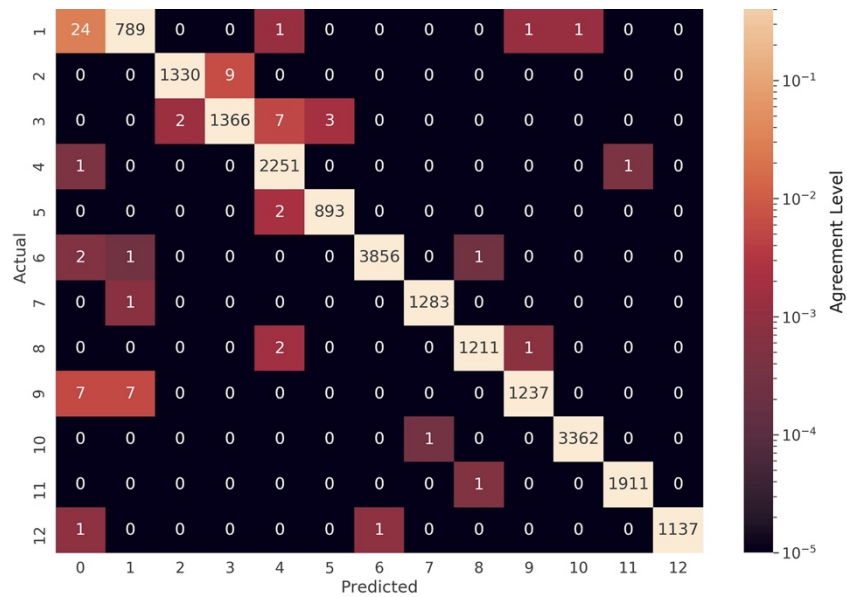
Methods		Measurement (%)				
		Accuracy	Precision	Recall	F1-Score	Images/s
Binary Classification	MCellNetB	99.77	99.53	98.66	99.09	343
	MCellNetA	99.64	98.71	98.44	98.57	346
	MNetSearch	99.51	98.08	98.12	98.10	320
	MobileNetV2	99.47	98.15	97.70	97.92	258
Multiclass Classification	MCellNetM	99.69	99.67	99.41	99.54	343
	MobileNetV2	99.59	99.39	99.41	99.40	251

3.3.3 Multiclass Classification

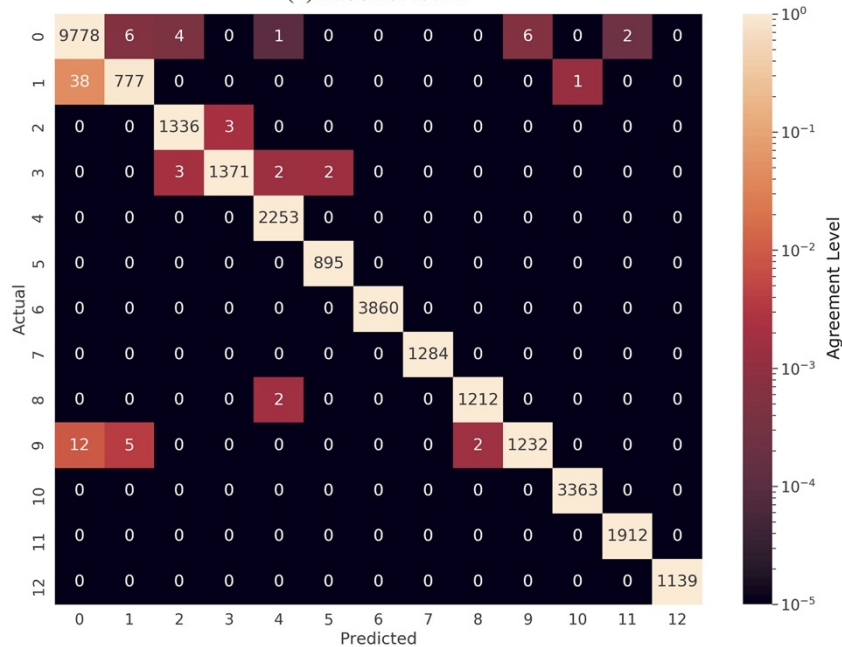
MCellNet multiclass classification model (MCellNetM) shows a comparable performance in terms of average accuracy, precision, recall and F1-scores as compared to MobileNetV2 (**Table 3-4**). The individual precision and recall score on multiclass classification are shown in **Table 3-5**.

The accuracy of MCellNetM reaches 99.69% using the macro average [155]. In terms of the processing speed, MCellNetM is about 37% faster (343 frames per

second) than MobileNetV2 (251 frames per second) in multiclass classification task on Nvidia Jetson TX2 board. The confusion matrices for MobileNetV2 and MCellNetM are shown in **Figures 3-11a** and **b**, respectively. In an ideal case with no misclassification, all boxes along the diagonal from the top left to bottom right have a value of 1 and appear white, and all the rest boxes have a value of zero and appear black. With misclassification, the non-diagonal boxes will have non-zero values and appear red. As seen in **Figure 3-11**, the confusion matrix of MobileNetV2 has significantly more red boxes, indicating higher classification errors. MobileNetV2 has a less satisfactory performance even in the classification of microplastics which are supposedly easier to classify compared to *Cryptosporidium* and *Giardia*. This outcome may possibly result from overfitting of MobileNetV2 for it has richer parameters than MCellNetM. In MCellNetM, the microplastics are well separated. In very few occasions, misclassification is observed for microplastics of similar size. For example, a few 3- μm microbeads are misidentified as 1.54- μm and 4- μm microbeads. This error possibly results from a few out-of-focus images under existing autofocusing mechanism. In such a scenario, smaller objects may appear bigger. Another possible source of error may come from the natural water pollutants which come in diverse sizes and shapes. Some of the particulate pollutants show image patterns similar to bioparticles, leading to misclassification of pollutants to *Cryptosporidium* and *Giardia*.



(a) MobileNetV2



(b) MCellNetM

0: Pollutants 1: *Cryptosporidium* 2: 1.54 μ m beads 3: 3 μ m beads 4: 4 μ m beads
 5: 4.6 μ m beads 6: 5 μ m beads 7: 5.64 μ m beads 8: 8 μ m beads 9: *Giardia*
 10: 10 μ m beads 11: 12 μ m beads 12: 15 μ m beads

Figure 3-11: The normalized confusion matrix on multiclass classification task. (a) MobileNetV2 and (b) Our proposed MCellNetM. 13 classes of output include naturally existing pollutants, *Cryptosporidium*, *Giardia*, and beads: 1.54 μ m, 3 μ m, 4 μ m, 4.6 μ m, 5 μ m, 5.64 μ m, 8 μ m, 10 μ m, 12 μ m, and 15 μ m. The color grading represents the level of agreement between the prediction and the ground truth with white (1) being full agreement and black (0) being no agreement. In an ideal case with no misclassification, all boxes along the diagonal from the top left to bottom

right have a value of 1 and appear white, and all the rest boxes have a value of zero and appear black. With misclassification, the non-diagonal boxes would have non-zero values and appear red.

The feature maps generated from the IRB0-5 of MCellNetM in multiclass classification task are also examined with the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm [156]. t-SNE is a widely adopted visualization method that projects high-dimensional data into low dimensions and visualize in low dimensions. t-SNE gives you the intuition on how the data is arranged in high-dimensional space. Unlike Principal Components Analysis (PCA) [157], t-SNE is a non-linear dimensions reduction technique. The PCA is a linear dimensions reduction technique that target to maximum variance and preserves large pairwise distances, and the t-SNE only preserves small pairwise distances (local similarities). The t-SNE algorithm builds the similarity relationship between the pairs of instances in high dimensional and low dimensional spaces.

As shown in **Figure 3-12**, points in different colors that represent respective classes are projected from the vector presentation of respective IRB outputs into a two-dimensional space. The t-SNE graphs from IRB0 to IRB5 (a-f) show that the microbeads are well separated even in shallow layers. But deeper layers for the feature transformation are required to distinguish *Cryptosporidium* and *Giardia* from each other. These results suggest that the structure and the size of the neural network must match the task in terms of data size and distribution in order to balance the accuracy and speed.

Table 3-5: Individual precision and recall score on multiple classification

Class	MCellNetM		MobileNetV2	
	Precision	Recall	Precision	Recall
Pollutants	99.51	99.79	99.67	99.55
<i>Cryptosporidium</i>	98.14	95.07	95.36	96.51
1.54 μm	99.40	99.78	99.85	99.33
3 μm	99.78	99.49	99.35	99.13
4 μm	99.78	100.0	99.34	99.91
4.6 μm	99.78	100.0	99.67	99.78
5 μm	100.0	100.0	99.59	99.90
5.64 μm	100.0	100.0	99.92	99.92
8 μm	99.84	99.84	99.75	99.75
<i>Giardia</i>	99.56	98.40	99.63	98.84
10 μm	99.97	100.0	99.97	99.97
12 μm	99.90	100.0	99.95	99.95
15 μm	100.0	100.0	100.0	99.82
Avg	99.67	99.41	99.39	99.41

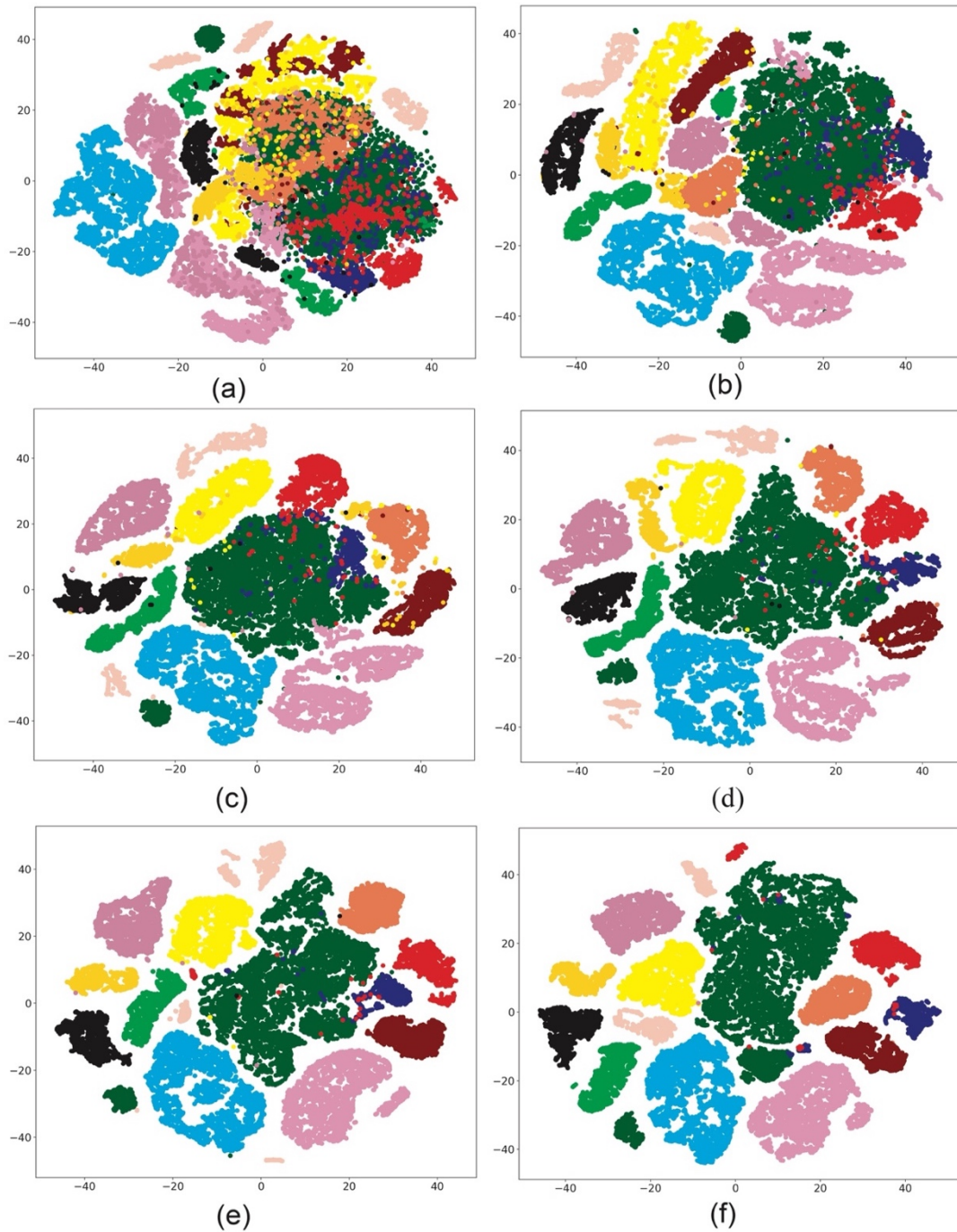


Figure 3-12: The t-SNE visualization on the IRB0 to IRB5 (a)-(f) of MCellNetM in multiclass classification. Different color is the different target class of *Cryptosporidium*, *Giardia*, microbeads (1.54 μm , 3 μm , 4 μm , 4.6 μm , 5 μm , 5.64 μm , 8 μm , 10 μm , 12 μm , and 15 μm .), and natural pollutants.

3.3.4 Binary Classification

Table 3-6: Error rate of different approaches on binary classification

Methods	EER (%)	0.1%FNR (%)	0.1%FPR (%)
MCellNetB	0.63	20.73	2.12
MCellNetA	0.73	4.19	5.34
MNetSearch	1.18	11.19	7.89
MobileNetV2	1.45	39.87	7.29

MCellNet binary classification model is used to evaluate the performance on two-class classification condition. Results indicate that MCellNetA and MCellNetB are superior to MobileNetV2 in terms of average accuracy, precision, recall and F1-score (**Table 3-4**). In terms of the processing speed, MCellNetA achieves 346 frames per second and surpasses MobileNetV2 (258 frames per second) by 34% in binary classification task on Nvidia Jetson TX2 board. **Figure 3-13** shows the False Negative Rate-False Positive Rate (FNR-FPR) curve of MCellNetA and MCellNetB compared to MobileNetV2 and MNetSearch. MCellNetA and MCellNetB have significantly better performances with improved error rate compared to MobileNetV2, a well-established deep neural network architecture. The Equal Error Rate (EER), FNR at $FPR = 0.1\%$ and FPR at $FNR = 0.1\%$ are summarized in **Table 3-6**. MCellNetB has an error rate of only 0.63% at EER. In comparison, MobileNetV2 has a relatively large error rate of 1.45%. At the false negative rate of 0.1%, MCellNetA has a false-positive rate of 4.19%, whereas it is 39.87% for MobileNetV2. At the false positive rate of 0.1%, the false-negative rate of MCellNetB is 2.12%, whereas it is 7.29% for MobileNetV2 model.

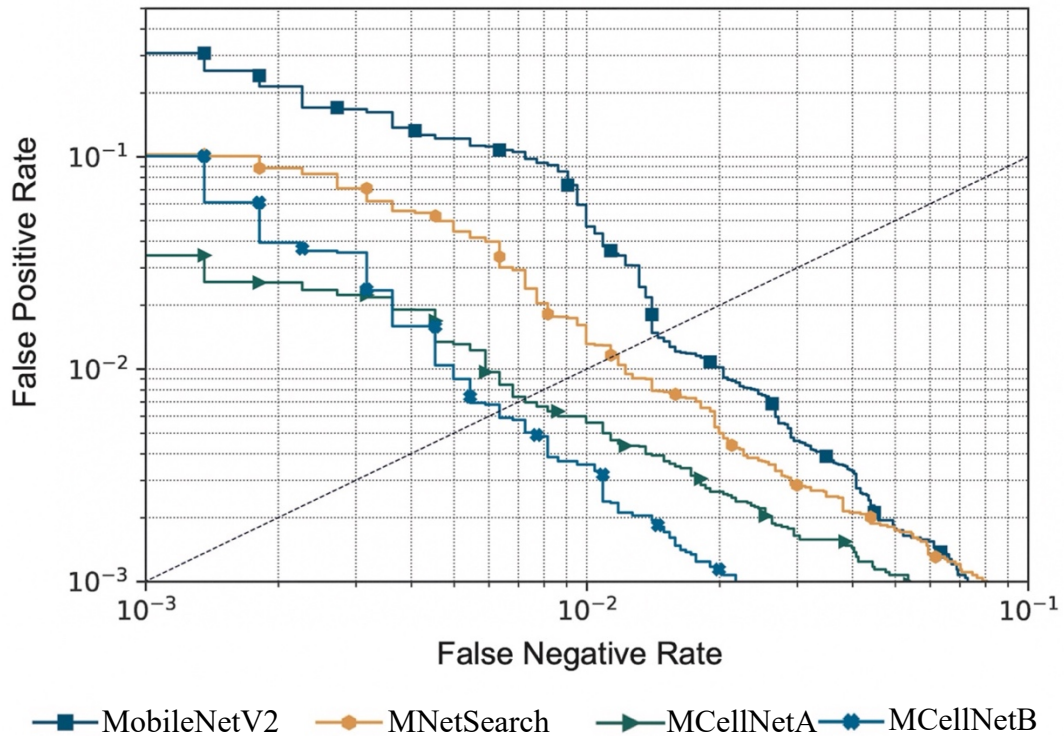


Figure 3-13: The False Negative-False Positive rate curve on the binary classification task. The diagonal dashed is the EER. Natural pollutant images are class 0, and all *Cryptosporidium* and *Giardia* images are class 1.

MCellNetB is evolved from multiclass classifier through a multiclass-to-binary strategy. MCellNetB is first trained with multiple outputs, 13 classes for our case. Next, binary class prediction labels are generated by merging the outputs into two classes. For instance, all labels for microplastics and natural pollutants are merged into a new pollutant class label "0", and both *Cryptosporidium* and *Giardia* labels are merged into a protozoa class "1". Using this approach, MCellNetB has achieved an accuracy of 99.77%. Results show that the multiclass-to-binary classifier MCellNetB outperforms the binary classifier (MCellNetA) and other deep neural networks in the binary classification task in terms of average accuracy, precision, recall and F1-score (Table 3-4). This observation suggests that rich labeling may offer more hints for optimization, thereby generating more precise filters for higher accuracies [158]. In term of protozoa classification, the MCellNetB achieves a sensitivity of 97.37%, a

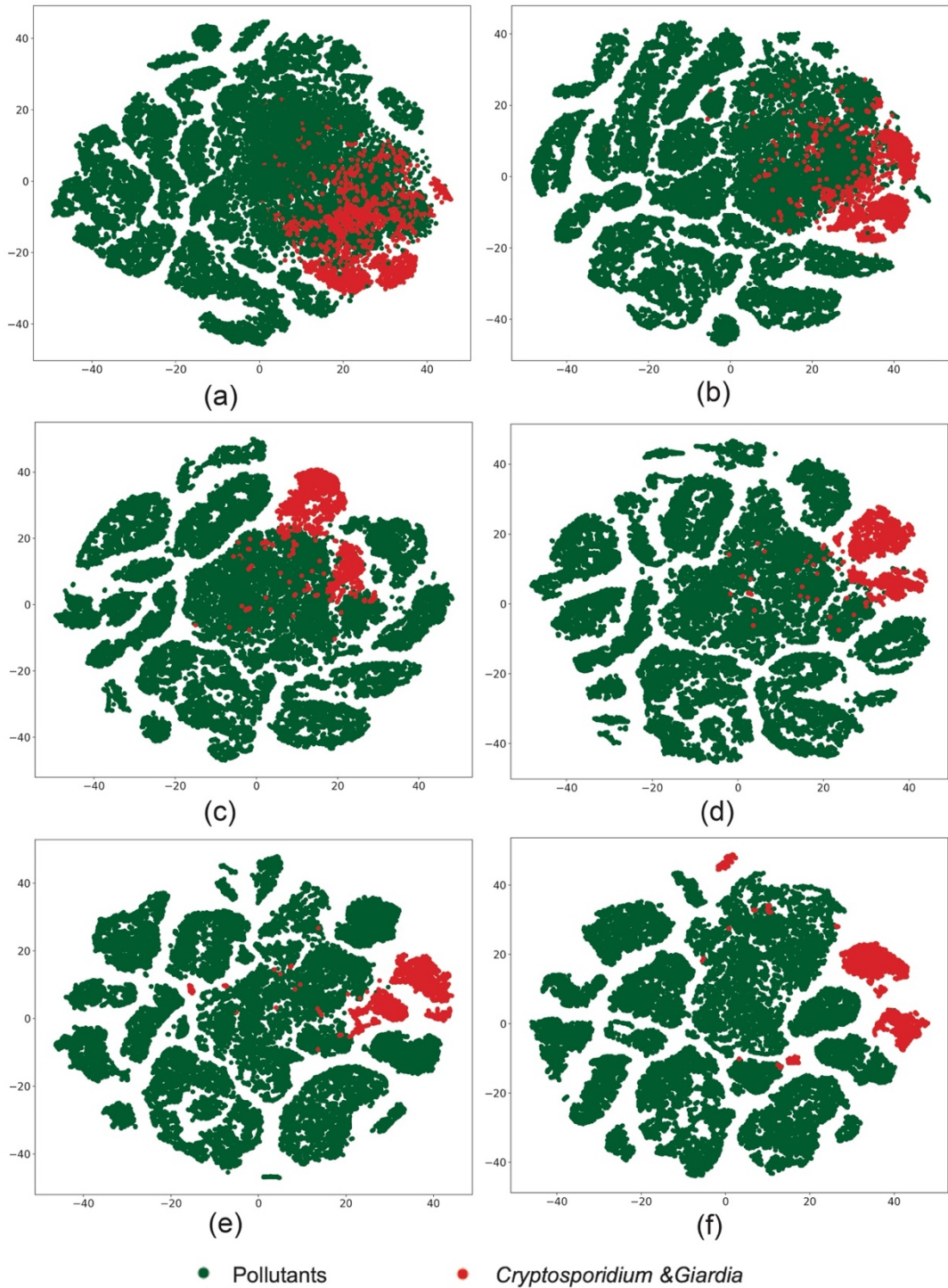


Figure 3-14: The t-SNE visualization on the IRB0 to IRB5 (a)-(f) of MCellNetB in binary classification. Natural pollutant images are class 0 (green), and all *Cryptosporidium* and *Giardia* images are class 1 (red).

specificity of 99.95%, a positive predictive value of 99.26% and a negative predictive value of 99.92%.

The feature maps generated from the IRB0-5 of MCellNetB in the binary classification task with the t-SNE algorithm were also examined. As shown in **Figure 3-14**, points in green represent microplastics and natural pollutants, and points in red represent protozoans. The t-SNE graphs from IRB0 to IRB5 (**a-f**) show that the microplastics and pollutants are well separated even in the shallow layers.

3.4 Summary

In this Chapter, a deep neural network is demonstrated by using imaging flow cytometry as a system for the *Cryptosporidium* and *Giardia* detection. The self-developed deep neural network MCellNet adopts the building block (IRB) from MobileNetV2 to achieve faster speed and lower power consumption for affordable machines. We used well-characterized samples, such as commercial *Cryptosporidium* and *Giardia* cell line and microplastic beads to demonstrate the capability of the proposed approach. For multiclass and binary classifications, MCellNet achieves accuracies of 99.69% and 99.7%, respectively. Our system is able to detect *Cryptosporidium* and *Giardia* with a sensitivity of 97.37% and a specificity of 99.95%. MCellNet has an analysis speed of 346 frames per second outperforming the state-of-the-art deep learning algorithm MobileNetV2 (251 frames per second) in demonstrated tasks. We also notice that even with good training, the machine learning model may still not work well on novelty data that the model did not see before. Deep learning with Bayesian inference may be a good solution. With enough high-quality training data, MCellNet could be extended to detect other types of bioparticles in high speed. The reported system, with its ability to detect and classify *Cryptosporidium* and *Giardia* with high speed and accuracy, would empower water monitoring and enable users to determine high-risk bio-contaminants and low-risk microplastics. It could also be potentially applied to other high-throughput single-cell analysis applications for environmental monitoring, clinical diagnostics, and other biomedical fields.

Chapter 4 Deep Metric Learning-Enabled Rare Bioparticle Detection

Recently, deep neural networks have shown superb performance in analysing bioimages for diseases diagnosis and bioparticle classification. Conventional deep neural networks use simple classifiers such as SoftMax to obtain high accuracy results. Therefore, they have limitations in many practical applications which require both low false alarm rate and high recovery, e.g., rare bioparticle detection, in which the representative image data is hard to collect, the training data is imbalance, and the input images in inference time could be different from the training images. Deep metric learning offers a better generatability by using distance information to model the similarity of the images and learning a function maps from images pixels to latent space, playing a vital role in the rare object detection. In this paper, we propose a robust model based on deep metric neural network for rare bioparticle detection. Experimental results show that the deep metric neural network achieved a high accuracy of 99.86% in classification, 98.89% in precision rate, 99.16% in recall rate and zero false alarm rate. The reported model empowers the imaging flow cytometry with capabilities of biomedical diagnosis, environmental monitoring, and other biosensing applications.

4.1 Introduction

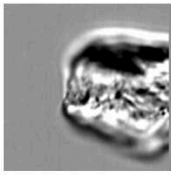



Pollution				
Detection Result	<i>Giardia</i>	<i>Giardia</i>	<i>Giardia</i>	<i>Cryptosporidium</i>
Confidence Level	0.99999	0.99999	0.99673	0.99988

Figure 4-1: Novel examples in the inference time. The conventional deep neural networks often make wrong predictions and do so confidently on some novel examples when the images are not seen in the training dataset. Such as the pollutions are predicted as *Giardia* or *Cryptosporidium* with confidence level $> 99.99\%$.

Rare bioparticle detection is essential to various applications such as cancer diagnosis and prognosis, viral infections, and implementing early warning systems [4, 5, 25, 36-39]. In those applications, the target bioparticles in the sample are extremely rare with a huge abundant of background particles. For example, the ratio of the target bioparticle and background bioparticles could be 1 in 1000 (0.1%) or even less. Currently, bioimage analysis has made a huge progress, benefitting from rich-dataset supervised learning using deep neural networks [4, 5, 39]. However, conventional deep neural networks only use simple classifiers such as SoftMax to obtain high accuracy results with the confidence that the deep neural network learns more distinct features than traditional machine learning in classification. Thus, in many practical applications, e.g., rare bioparticle detection, they sometimes get unexpected results [41, 90, 159, 160]. This is because it is hard to collect representative image data in those applications and the input images in inference time may be distinct from those during training. Those applications also require the model to have a performance of low false alarm as well as high recovery rate in practical environments. For example,

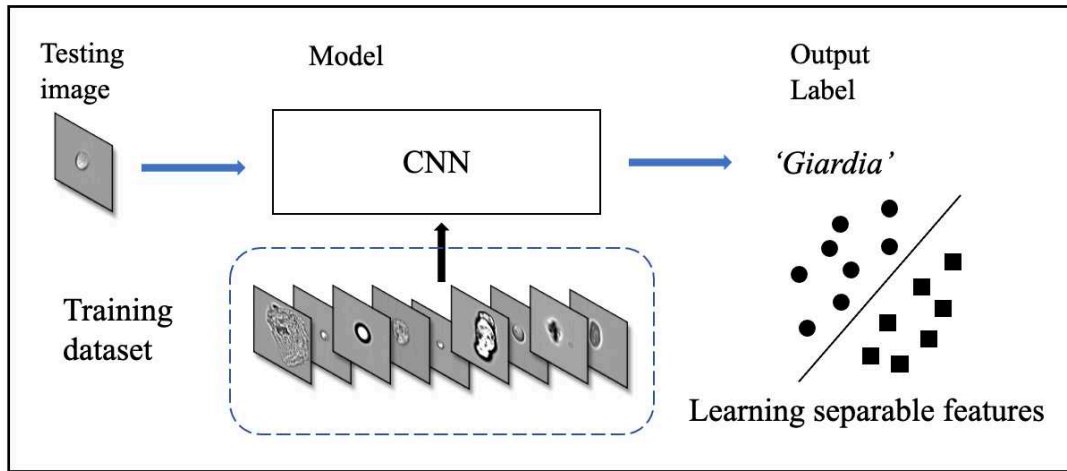


Figure 4-2: Conventional deep classification. In deep classification, the model only studies a boundary.

a large amount of false alarms will introduce high-cost consequential actions [161]. Up to now, it remains a great challenge in the detection of rare bioparticles in practical applications.

Conventional deep neural networks use simple classifier to make the decision of seen/unseen classes. Therefore, they often make wrong predictions, and do so confidently [162-165]. For example, the conventional deep neural network model sometimes predicts wrongly (It predicts the pollutants as *Cryptosporidium* or *Giardia*) with a high confidence level ($> 99.99\%$) as shown in **Figure 4-1**. These inaccuracies arise from the conventional classification approaches (**Figure 4-2**), for example, convolutional neural networks (CNNs) use Softmax [160] classifier, which acts as a linear classifier, and limit their ability to detect novel examples [41, 162, 164, 166, 167]. As a result, conventional Softmax based approaches are not suitable for open-set rare bioparticle detection. For example, a highly accurate algorithm based on a sophisticated densely connected neural network for bioparticle classification was developed for rare bioparticle detection [5], but it only achieved a sensitivity and specificity of 77.3% and 99.5%, respectively.

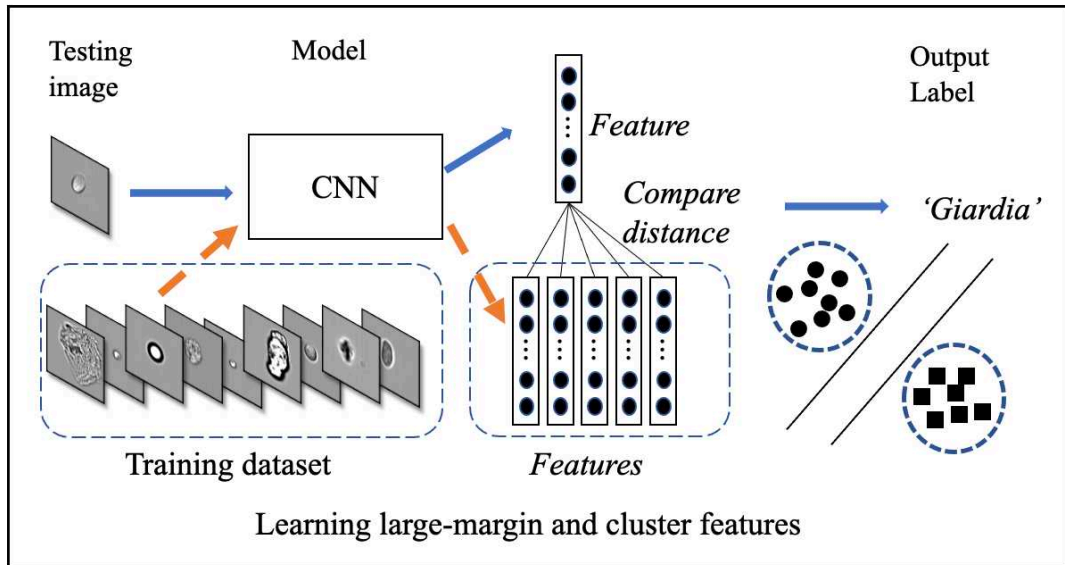


Figure 4-3: Deep metric learning problem. In deep metric learning, the model studies a more generative representation with similar classes are close and the unsimilar classes are far away.

Deep metric learning [90] (**Figure 4-3**) provides a possible direction to improve out of distribution or open-set detection by learning a map from the input image space to an output embedding features in the latent space. Instead of using SoftMax classifier, this approach uses semantic similarity such as the Euclidean distance to constrain the models. It does not rely on cross-entropy loss but proposes another class of network loss, i.e., contrastive loss. Thus, the sum of the output class probabilities is not to be one and this provides it a generatability [41]. Generative model is essentially a metric learning problem whereby the key is to learn a large margin distance metric within the latent space when the testing data are usually disjoint from the training dataset.

In this Chapter, a deep neural network model based on deep metric learning for rare bioparticle detection is demonstrated. The model leverages convolutional neural network to study the rich features in the dataset and learning distinct metric by using Siamese network [168] and contrastive loss, which maximizes the distance of

different classes and minimizes the distance of similar classes. Finally, experimental results such as comparing with conventional deep learning and rare bioparticle detection are presented.

4.2 Model for Rare Bioparticle Detection

4.2.1 The Fundamental of Deep Metric Learning

Unsupervised deep metric learning is used to learn a low-dimensional subspace to preserve useful geometrical information of samples. On the other hand, supervised deep metric learning is used to learn a projection from the sample space to the feature space and measure the Euclidean metric in this feature space to discriminate the results. The metric learning is defined to study a map function f with a dataset $\mathcal{X} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots\}$, whereby $f: \mathcal{X} \rightarrow \mathbb{R}^n$ is well defined mapping and $d: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ is the Euclidean distance over \mathbb{R}^n . Then, $d_f(\mathbf{x}, \mathbf{y}) = d(f(\mathbf{x}), f(\mathbf{y})) = \|f(\mathbf{x}) - f(\mathbf{y})\|_2$ is close to zero when \mathbf{x} and \mathbf{y} are similar. The mathematical definition of Euclidean distance $d(\mathbf{x}, \mathbf{y})$ is expressed as [90]

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \|f(\mathbf{x}) - f(\mathbf{y})\|_2 \\ &= \sqrt{(f(\mathbf{x}) - f(\mathbf{y}))^T (f(\mathbf{x}) - f(\mathbf{y}))} \end{aligned} \quad (4-1)$$

where $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and it is assumed that metric $d(\mathbf{x}, \mathbf{y}): \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ satisfies the following properties as

$$d(\mathbf{x}, \mathbf{y}) \geq 0 \quad (4-2a)$$

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \quad (4-2b)$$

$$d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \quad (4-2c)$$

and

$$d(x, x) = 0 \quad (4-2d)$$

Deep metric learning is widely applied in signature verification [168], face verification and recognition [169], and person re-identification [170].

4.2.2 Deep Metric Learning for Rare Bioparticle Detection

The embedding network structure of deep metric learning model is shown in **Figure 4-4**. The input of base network is a grayscale image with 120×120 pixels and one convolutional layer with a filter size of 7×7 is in the first stage. Then, three residual network blocks (RB0 to RB2) [81] are attached to the first convolution neural network layer. The output of the last residual network block RB2 is flattened, and then followed by a fully-connected layer [136] together with a parametric ReLU (PReLU) layer [171]. Finally, a Fully-Connected layer with 2 output units is attached to the PReLU layer to generate the latent feature vector of bioparticles. The detail parameters of the embedding network are listed in **Table 4-1**.

Table 4-1: Network parameters of the embedding network

Layer/Block	Type	Output dimension	Params
Conv2d	Convolution	$60 \times 60 \times 64$	9536
RB0	Residual Block	$30 \times 30 \times 64$	147,968
RB1	Residual Block	$15 \times 15 \times 128$	525,568
RB2	Residual Block	$8 \times 8 \times 256$	2,099,712
Pool	Average Pool	$2 \times 2 \times 256$	0
Dense1	Fully Connected	256	262,400
PReLU	Parametric ReLU	256	1
Dense2	Fully Connected	2	514

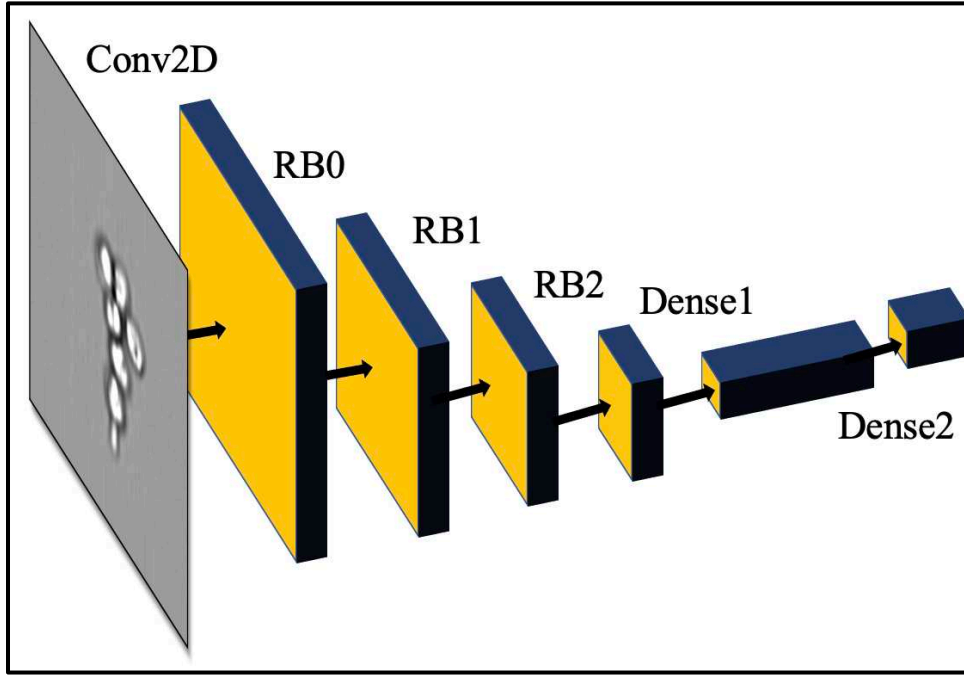


Figure 4-4: Embedding network structure.

The first convolutional 2D layer (Conv2D in **Figure 4-4**) [146] takes an $h \times w \times n$ input feature map \mathbf{X}^i , where h is the spatial height, w is the spatial width and n is the output channels of the feature map ($120 \times 120 \times 1$).

The input \mathbf{X}^i is transformed into a $60 \times 60 \times 64$ output feature maps \mathbf{X}^o and expressed as [86]

$$\mathbf{X}_{x,y,z}^o = \delta\left(\sum_{i,j,k} \mathbf{F}_{i,j,k,z} \cdot \mathbf{X}_{x+i-1,y+j-1,k}^i + \mathbf{b}_z\right) \quad (4-3)$$

where $z = 1, 2, \dots, m$ and $k = 1, 2, \dots, n$. The input feature map \mathbf{X}^i is convoluted with a number of feature detectors, each of which is a three-dimensional filter \mathbf{F} in the present layer ($7 \times 7 \times 1$), and a bias \mathbf{b} . An ReLU function $\delta(x)$ is attached to this convolution operator.

Three cascaded residual network blocks (RB0-2 in **Figure 4-4**) [81] with down sample (stride = (2, 2)) are attached to the first convolution layer. The RB has two 3×3 convolutional layers and the same number of output channels as shown in **Figure 4-5**. In the end, a batch normalization layer and a ReLU activation function

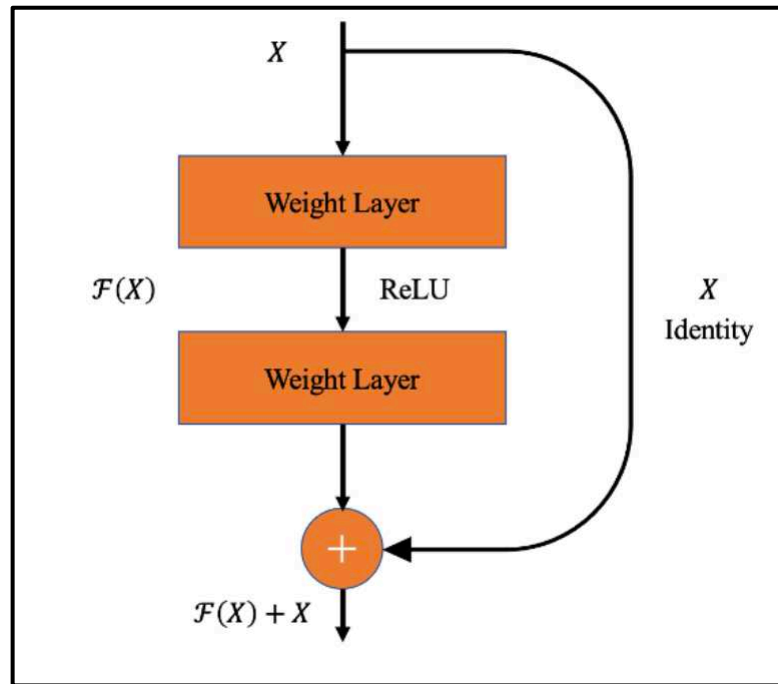


Figure 4-5: Residual blocks in the embedding network.

follow each convolutional layer. In addition, an identify path is added to connect the input to the output directly.

The classifier is implemented by two fully connected layers (Dense Layer in **Figure 4-4**). It takes the last output of RB2 as the input and applied cascaded matrix multiplications and non-linear function to the weight matrix \mathbf{F} and bias \mathbf{b} to produce a vector with two dimensions in the latent space. The equation of fully-connected layer can be expressed as

$$\mathbf{X}^o = \delta(\mathbf{F}\mathbf{X}^i + \mathbf{b}) \quad (4-4)$$

Parametric ReLU (PReLU) layer is used after the fully-connected layer. The parametric ReLU layer is expressed as

$$f(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ a_i x_i, & \text{if } x_i \leq 0 \end{cases} \quad (4-5)$$

where x_i is the input value and a_i is the parameter of the PReLU layer.

4.2.3 Training of Deep Metric Learning

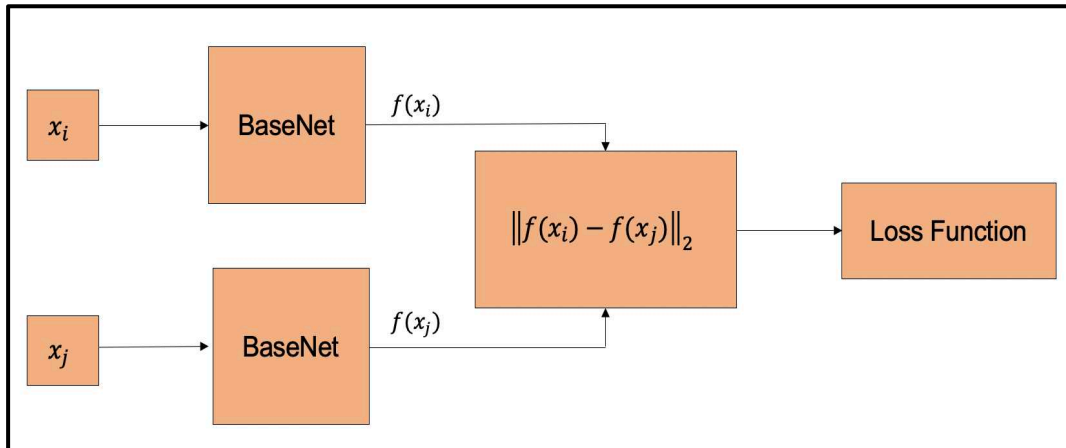


Figure 4-6: The structure of Siamese network for training deep metric learning. The twin networks share same network parameters. A loss function is attached to this twin network to regularize the network.

Siamese network [168] is the most popular deep metric learning network structure which is used to train the deep learning model as shown in **Figure 4-6**. The network consists of two embedding networks and a joint output neuron. In this work, residual network blocks are used as embedding networks to extract features. The two embedding networks share the same weights, the identical sub-networks extract feature vectors from two images simultaneously and the joined neuron measures the distance between the two feature vectors in the latent space using a metric. In the training process, the similar and dissimilar pairs (x_i and x_j) are passed through the network and generate features vector in the latent space named $f(x_i)$ and $f(x_j)$. In the loss function, the distance metric $d(x, y) = \|f(x) - f(y)\|_2$ is regressed to minimize the distance between similar pairs and keep the distance of the dissimilar pairs. The contrastive loss is used to train the Siamese network. For the pair of input (x_i, x_j) , it is a positive pair if x_i and x_j are semantically similar and negative pair if they are dissimilar. The training process of Siamese network deals with minimize a

contrastive loss which is expressed as

$$\begin{aligned}
 & L\left(\{W^{(m)}, b^{(m)}\}_{m=1}^M\right) \\
 &= \sum_{(i,j) \in \mathcal{S}} h(d_f(\mathbf{x}_i, \mathbf{x}_j) - \tau_1)^2 + \sum_{(i,j) \in \mathcal{D}} h(\tau_2 - d_f(\mathbf{x}_i, \mathbf{x}_j))^2
 \end{aligned} \tag{4-6}$$

where $h(x) = \max(0, x)$ is the hinge loss function, $\tau_1 = 0.9$ and $\tau_2 = 1.0$ are two positive thresholds with $\tau_1 < \tau_2$, respectively, and $\mathcal{S} = \{(i, j)\}$ is the similar pairs and $\mathcal{D} = \{(i, j)\}$ is the dissimilar pairs.

The deep metric learning model is implemented with deep learning framework-PyTorch [172] and trained over an Ubuntu GPU server [149] with four Nvidia GeForce RTX 2080 cards as well as the Intel Xeon CPU E5-2650. To train and evaluate the performance of the model, the selected image dataset is randomly split into the training, validation and testing dataset with 48%, 12% and 40% of the total number of images, respectively. Later, images in training dataset are augmented to 10,000 images, and each image is randomly sampled from the dataset and processed by position transformation, horizontal and vertical flipping, rotation or zooming. The weight of the deep neural networks is initialized with the Glorot uniform initializer [150] at a mean value of zero and a standard deviation at 10^{-2} , and the network is trained in an end-to-end fashion using the Adam stochastic optimizing algorithm [151]. The parameters for Adam are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate decay is used for training. Early stop is also used to prevent overfitting by stopping the training when the model's performance on validation dataset start to degrade [173]. A maximum of 300 epochs is used to train the model.

4.2.4 Deep Metric Learning based Model

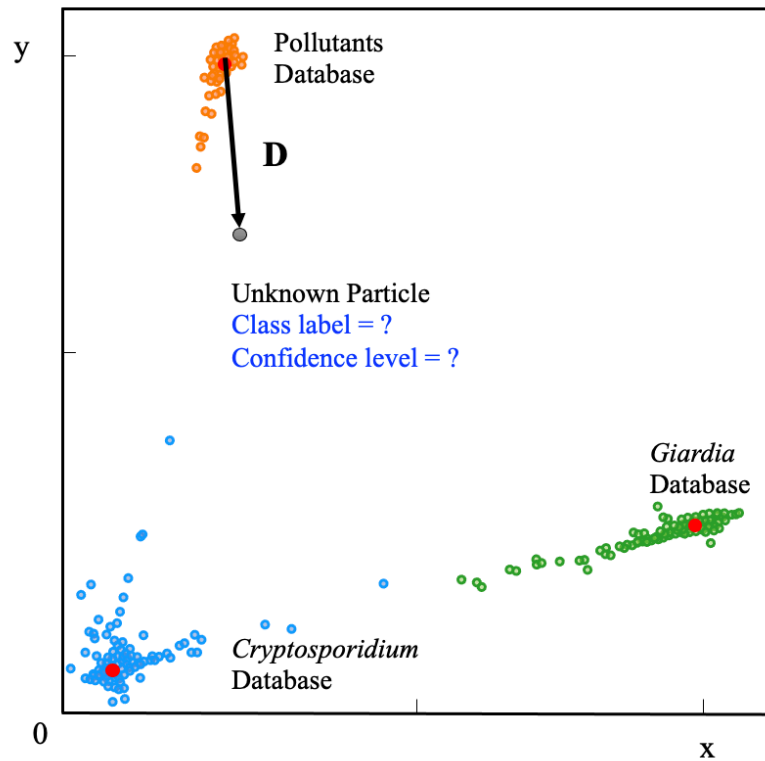


Figure 4-7: Deep metric learning based classification. The unknown bioparticle is classified to correspond classes. Class label is assigned to classify the unknown particles by the closed cluster center (red). Confidence level is used to present the similarity of unknown particles to certain databased collected by *Cryptosporidium* and *Giardia* samples.

The deep metric network studies a map from images into a latent space and it cannot be directly used to classify images. In order to classify the rare bioparticle images with deep metric learning model, further processing is needed in the end of this neural network model. It converts the values in the latent vector into a target class label and a confidence score. As shown in **Figure 4-7**, the class label is assigned by the closed cluster center, which can be calculated by either mean latent vectors (Mean center) or Gaussian Mixture Models (GMM) [174] of a known class, such as *Cryptosporidium*, in the training dataset. The confidence score is used to present the

similarity between the target bioparticle to the certain classes, which are collected in the training phase. The confidence score can be calculated by the distance of the target bioparticle to the center of certain class on the distribution diagram of the latent space or a Gaussian estimator.

Gaussian distribution [175] is a continuous probability distribution, which has a characteristic with symmetric “Bell curve” shape that quickly falls off toward 0. GMM is a probabilistic model, which assumes that the underlying data belong to a linear combination of several gaussian distributions. A GMM model gives a posterior distribution over K Gaussian distributions and show better performance on optimize model complexity [176]. The GMM can be represented as [177]

$$P(x|\pi, \mu, \Sigma) = \sum_{i=1}^K \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \quad (4-7)$$

where $\mathcal{N}(x|\mu, \Sigma)$ is a normal distribution, x is a multidimension vector variable, μ is the mean of this x and Σ is the covariance matrix. The $\mathcal{N}(x|\mu, \Sigma)$ is given by [177]

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (4-8)$$

where D is the number of dimensions of the feature vector. The π_i are mixing coefficients, satisfied $0 \leq \pi_i \leq 1$ and $\sum_{i=0}^N \pi_i = 1$. With the assumption that x_i are come from independent K mixture distributions insider C . The equation can be expressed as [177]

$$P(C|\pi, \mu, \Sigma) = \prod_{n=1}^N \sum_{i=1}^K \pi_i \mathcal{N}(x_n|\mu_i, \Sigma_i) \quad (4-9)$$

Expectation-maximization (EM) algorithm is used to find the local maximum likelihood and estimates of individual parameters in GMM (μ and Σ). EM is an iterative algorithm which follows the rule that very iteration strictly increases the maximum likelihood. EM algorithm may not approach to the global optimize point

but it can guarantee to local saddle point. The EM algorithm consists two main steps: expectation and maximization.

The expectation step calculates the expectation of the clusters when each $x_i \in X$ is assigned to the clusters with given μ, Σ, π . The maximization step maximizes the expectation in previous step by find suitable parameters.

First, the program randomly assigns samples $X = \{x_1, x_2, \dots, x_n\}$ to components estimated mean $\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_k$. For example, $\hat{\mu}_1 = x_6, \hat{\mu}_2 = x_{20}, \hat{\mu}_3 = x_{21}, \hat{\mu}_4 = x_{33}, \hat{\mu}_5 = x_{60}$ when $N = 100, K = 5$. And assign $\hat{\Sigma}_1 = \hat{\Sigma}_2 = \dots = \hat{\Sigma}_k = Cov(x) = E[(X - \bar{x})(X - \bar{x})^T]$ where $\bar{x} = E(X)$. Last, all the mixing coefficients are set to a uniform distribution with $\hat{\pi}_1 = \hat{\pi}_2 = \dots = \hat{\pi}_k = \frac{1}{K}$.

In the expectation step, the $p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)$ is given by [177]

$$p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k) = \frac{\hat{\pi}_k \mathcal{N}(\hat{x}_i | \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^K \hat{\pi}_j \mathcal{N}(x_i | \hat{\mu}_j, \hat{\Sigma}_j)} \quad (4-10)$$

In the maximization step, $(\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)^{(i+1)} = \arg \max_{\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k} p(C_k | x_i, (\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)^i)$

where each item can be expressed as [177]

$$\hat{\pi}_k = \sum_{i=1}^N \frac{p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)}{N} \quad (4-11a)$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^N p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k) x_i}{\sum_{i=1}^N p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)} \quad (4-11b)$$

and

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^N p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k) (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^N p(C_k | x_i, \hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k)} \quad (4-11c)$$

The whole EM process repeats iteratively until the EM algorithm converges to point and gives a maximum likelihood estimate for each $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k$. With the EM process, the parameters of individual Gaussian distributions can be estimated. The studied model can be used to predict the label and confidence of a unknown bioparticle.

4.3 Experimental Results and Discussions

4.3.1 Profiling of Bioparticles

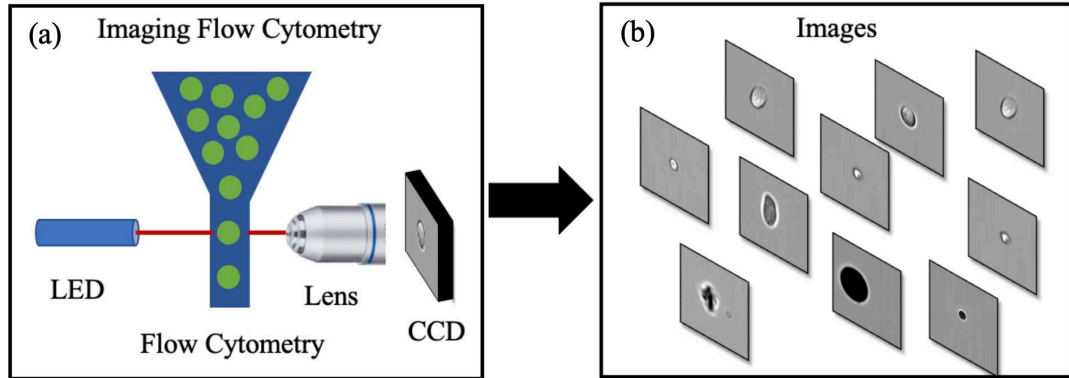


Figure 4-8: Overview of system setup. (a) A deep metric neural network that enables intelligent imaging flow cytometry (Amnis® ImageStream®X Mk II) for rare bioparticle detection. The system consists of laser, flow cytometry, imaging system, image database, and deep neural network, etc. (b) Images are collected into a database.

The rare bioparticle detection system is illustrated in **Figure 4-8**. First, samples were spiked and imaged using the imaging flow cytometry (Amnis® ImageStream®X Mk II). Bioparticles such as *Cryptosporidium*, *Giardia*, microplastics and other pollutants such as dirt and cell debris with size all from 3 to 14 μm that naturally exist in drinking water were included in the study. The naturally existing pollutants were obtained by concentrating 10 liters of drinking water using a water filtration system. Formalin-treated *Cryptosporidium* oocysts, *Giardia* cysts (Waterborne™ Inc) and synthetic microplastic beads (Thermo Fisher Scientific, Duke Scientific and Polysciences Inc.) of different sizes (1.54 μm , 3 μm , 4 μm , 4.6 μm , 5 μm , 5.64 μm , 8 μm , 10 μm , 12 μm and 15 μm) were spiked separately into 200 μL water. Bioparticles were hydrodynamically focused by a sheath flow and flowed through the detection region with PBS used as the sheath medium. Single bioparticles were illuminated with a LED light source, and bright-field images were acquired with a

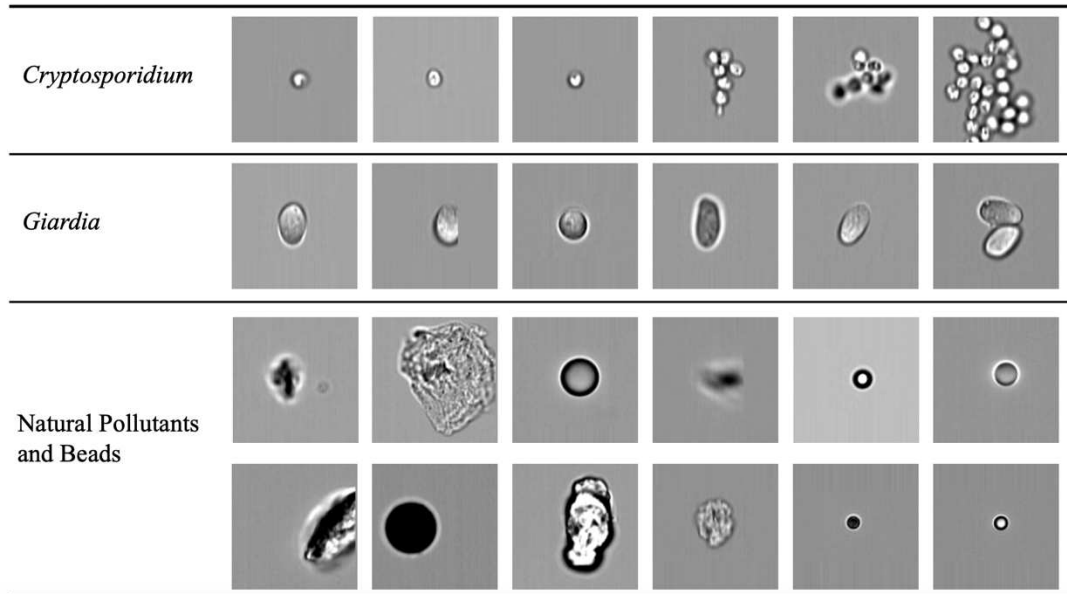


Figure 4-9: Bioparticle image dataset. Each row represents one type of bioparticle. From the top to bottom are *Cryptosporidium*, *Giardia*, natural pollutants and beads. All subfigures share the same scale bar.

CCD camera [142] using a $60\times$ objective (**Figure 4-8a**).

The raw image sequence files (.RIF) of different samples were captured. The raw bright-field images were extracted from the image sequence files by IDEAS software (accompanying with the ImageStream) and patched to 120×120 pixels (**Figure 4-8b**). From millions of raw images acquired, 89,663 images (**Figure 4-9**) were selected to construct the dataset by experts. The image dataset consists of three classes: *Cryptosporidium* (2,078 images), *Giardia* (3,438 images), and other natural pollutants and beads (84,147 images). The bright-field images of bioparticle had complex patterns, such as distinct sizes, degree of aggregation and different internal structures, which complicated the learning task.

4.3.2 Bioparticle Classification Evaluation

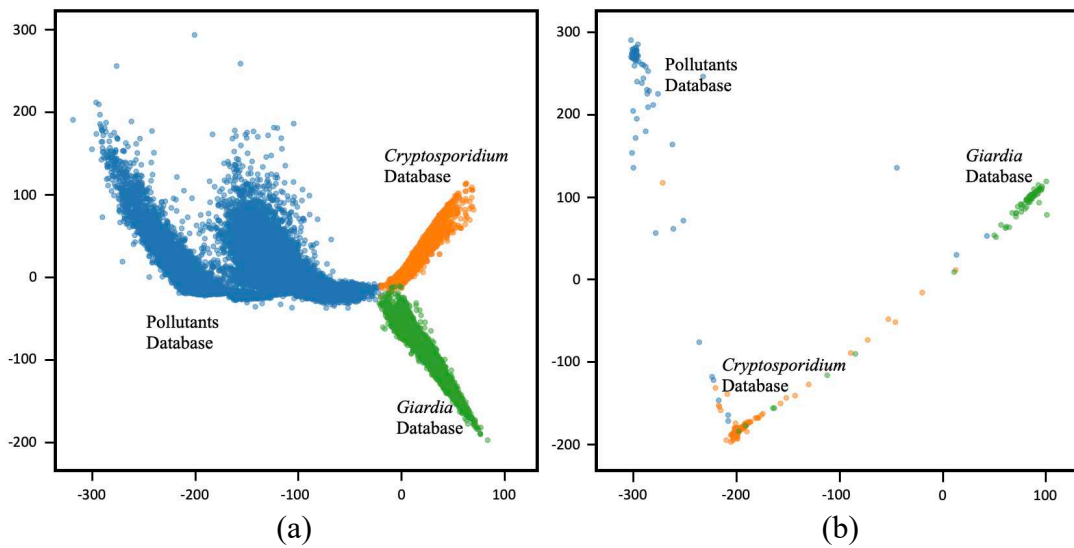


Figure 4-10: Visualization on 2D latent space of traditional deep classification-based model and deep metric learning. (a) traditional deep classification-based model, (b) deep metric learning based model.

The output latent vectors of the deep metric neural network, which are mapped to a 2D latent space as shown in **Figure 4-10**. Compared with the conventional deep classification method (**Figure 4-10a**), the deep metric learning model is trained using Siamese network (**Figure 4-10b**) and contrastive loss shows a better performance. The dots of the similar images in deep metric learning are closer and the dissimilar images are kept far away from others, providing the ability of generatability. Moreover, the t-SNE graphs from RB0 to RB2, which is low level to high level features (**Figure 4-11**), also show that the data is well separated in the deep metric learning based model (**Figure 4-11a,c, and e**) even in the shallow layers by comparing with the conventional classification-based method (**Figure 4-11b,d, and f**).

For the classification, the GMM model is selected because it can tell us how much confidence it is associated with the target cluster, and it has the same accuracy

of 99.86% with the mean center. The results of GMM with deep metric learning are shown in **Figure 4-12**. The blue color is the natural pollutants and beads, the yellow color is the *Cryptosporidium* and the green color is the *Giardia*. The dot and plus signs are training dataset and testing dataset respectively. The purple colour with white boundary is the estimated GMM cluster. The estimated parameters of GMM are listed in **Table 4-2**.

Table 4-2: Estimated parameter of GMM with deep metric learning

	Mean (μ)		Covariances (Σ)		Coefficients (π)
Nature	-0.4795	0.8864	3.9012E-6	-6.2244E-6	0.6390
Pollutants			-6.2244E-5	3.2312E-5	
<i>Cryptosporidium</i>	-0.2034	-0.7312	9.8237E-5	5.4069E-5	0.1806
			5.4069E-5	5.5438E-5	
<i>Giardia</i>	0.6001	0.5226	4.3410E-5	2.1701E-5	0.1804
			2.1701E-5	2.3743E-5	

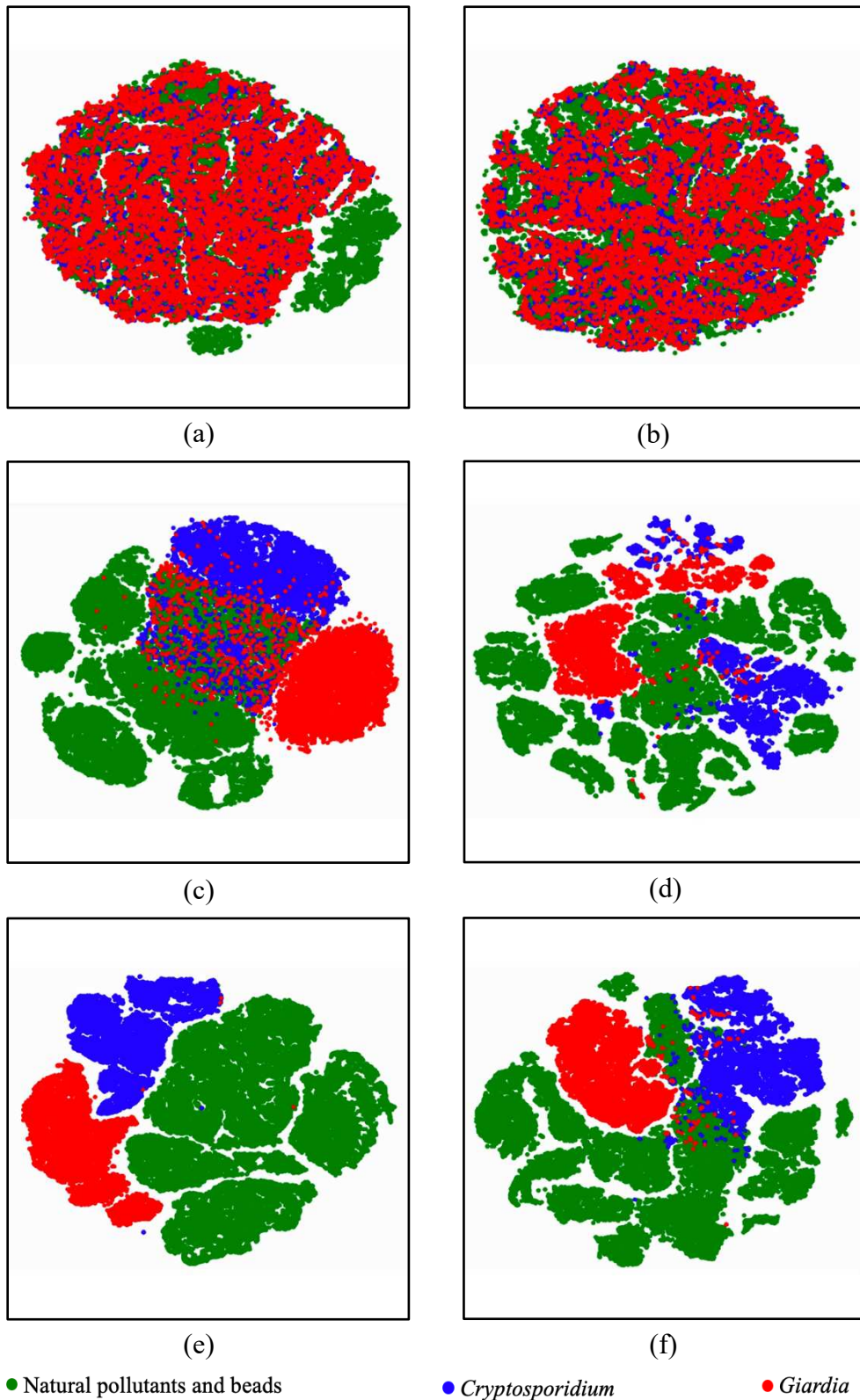


Figure 4-11: Visualization on intermediate layers with t-SNE on deep metric learning and traditional classification-based model. (a, c, e) the lower, middle, and high level of deep metric learning, (b, d, f) the lower, middle, and high level of traditional deep classification-based model.

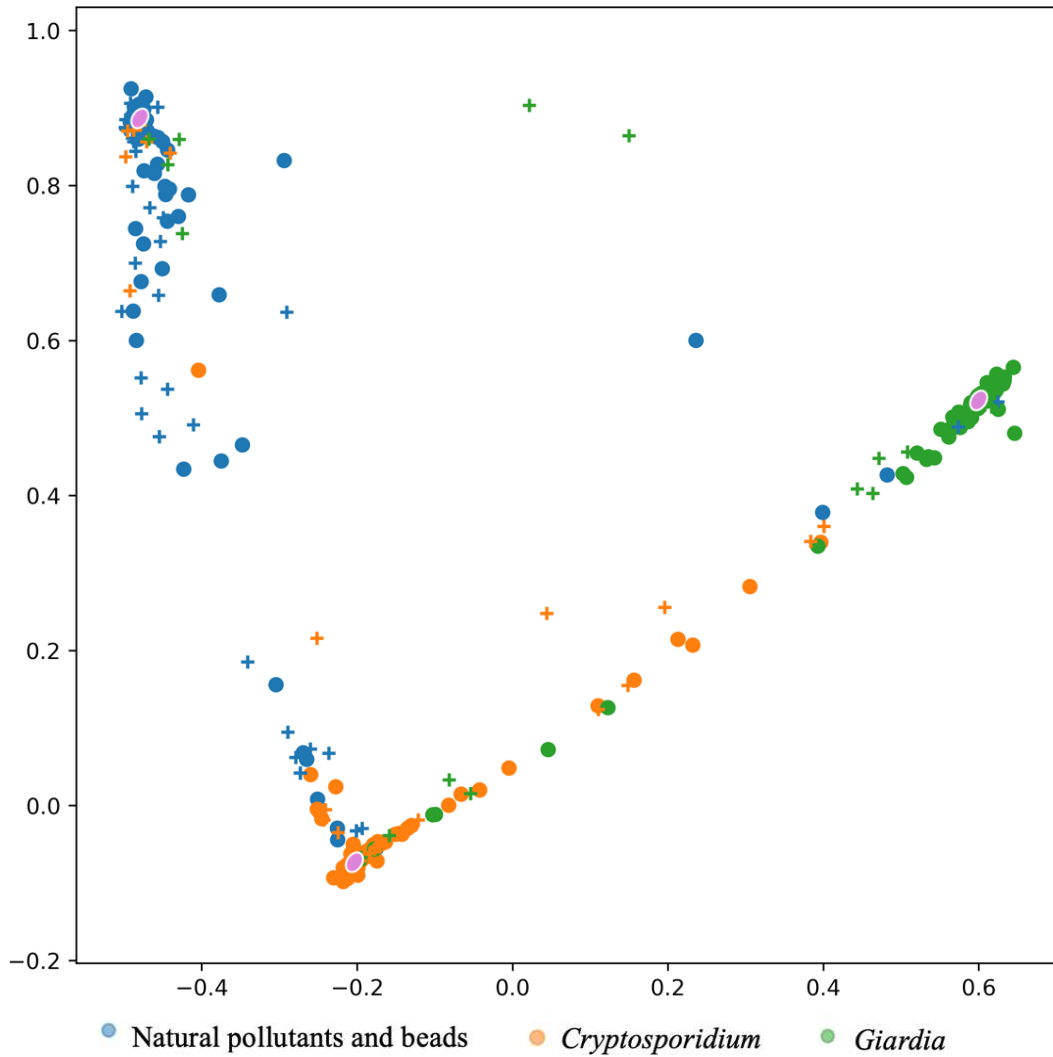


Figure 4-12: Visualize of the GMM result with deep metric learning. Blue color is the natural pollutants and beads, the yellow color is the *Cryptosporidium* and the green color is the *Giardia*. The purple colour with white boundary is the estimated GMM cluster.

The results of the model comparison between deep metric learning and conventional deep classification are summarized in **Table 4-3**. The model based on deep metric learning is superior to the model based on conventional deep classification neural networks in terms of accuracy, precision, recall and F1 score. The model based on deep metric learning network achieves 99.86% in accuracy, 98.89% in precision rate, 99.16% in recall rate and 99.02% in F1 score. On the other

hand, the model based on conventional deep classification gives 99.71% in accuracy, 97.84% in precision, 98.55% in recall and 98.19% in F1 score. The results in **Tables 4-4** and **4-5** also show that the performance of the individual class in deep metric learn-based model is better when it has a large quantity of training data. For example, nature contaminated particles.

Table 4-3: Precision, recall and F1-score on test dataset

Methods	Measurement (%)			
	Accuracy	Precision	Recall	F1 Score
Deep Classification	99.71	97.84	98.55	98.19
Deep Metric Learning	99.86	98.84	99.17	99.00

4.3.3 Model Verification using Spiked Samples

In order to evaluate the performance of the deep metric learning model on rare bioparticle detection, *Cryptosporidium* and *Giardia* were spiked into the concentrated water sample to emulate rare bioparticle in contaminated water. In total, ten tests were run, and the captured images were detected by the software with confidence level at 0.98 and verified by biological experts based on their morphologies. The results are summarized in **Table 4-6**. As shown in the table, the deep metric learning gives zero false warning signal, which is vital to implement the early warning system that needs the specificity of 100%. In comparing, conventional deep classification gives false positive signal in test 1, 3, 5, 6, 7, 10, especially false warning signals in test 7 and 10 are not acceptable. For the recovery rate, the deep metric learning gives an average of 85.5%.

Table 4-4: Confusion matrix of traditional deep classification

Class		Prediction		
		Nature Pollutants	<i>Cryptosporidium</i>	<i>Giardia</i>
Actual	Nature Pollutants	29610	35	14
	<i>Cryptosporidium</i>	20	807	4
	<i>Giardia</i>	10	8	1357

Table 4-5: Confusion matrix of deep metric learning-based classification

Class		Prediction		
		Nature Pollutants	<i>Cryptosporidium</i>	<i>Giardia</i>
Actual	Nature Pollutants	29639	17	3
	<i>Cryptosporidium</i>	9	820	2
	<i>Giardia</i>	6	9	1360

Table 4-6: *Cryptosporidium* and *Giardia* detection using deep metric learning

S/N	Spike Level	Images Number	Manual Counting	Sensitivity	Specificity	Alarm	Recovery Rate
1	20C	23483	7	85.7%	100%	Yes	85.7%
2	20C	18422	8	75.0%	100%	Yes	75.0%
3	20C	21834	10	80.0%	100%	Yes	80.0%
4	20G	19383	7	100.0%	100%	Yes	100.0%
5	20G	18320	9	88.9%	100%	Yes	88.9%
6	20G	24872	6	83.3%	100%	Yes	88.3%
7	0	20000	0	-	100%	No	-
8	0	20000	0	-	100%	No	-
9	0	20000	0	-	100%	No	-
10	0	20000	0	-	100%	No	-
Mean				85.5%	100%	-	85.5%

4.4 Summary

Siamese based deep metric learning provides a set of new tools for learning latent vectors by leveraging both convolutional neural network and deep metric learning. In this Chapter, a deep neural network based on deep metric learning for rare bioparticle detection is presented by incorporating Siamese constraint in the learning process. The model contains one convolutional layer with a filter size of 7×7 is in the first stage, three residual network blocks (RB0 to RB2) and two fully-connected layer together with a parametric ReLU (PReLU) layer with two output units to generate the latent feature vector of bioparticles. For the classification, the GMM model is selected because it can tell us how much confidence it is associated with the target cluster, and it has the same accuracy of 99.86% with the mean center. The model can learn interpretable latent representation that preserves semantic structure of similar and dissimilar images. Well-characterized samples, such as commercial *Cryptosporidium* and *Giardia* cell line and microplastic beads to demonstrate the capability of the proposed approach. The experimental results demonstrate that Siamese based deep metric learning can achieve classification-based accuracy while encoding more semantic structural information in the latent embedding and suit for rare bioparticle detection which achieved 99.86% in accuracy and zero false alarm with *Cryptosporidium* and *Giardia* under complex background images. With enough high-quality training data, the model could be extended to detect other types of rare bioparticles in high accuracy. The model would empower intelligent imaging flow cytometry for rare bioparticle detection in biomedical diagnosis, environmental monitoring, and other biosensing applications.

Chapter 5 Machine Learning-Based Pipeline for Bioparticle Sizing

High accuracy measurement of size of bioparticle imaging is essential in biomedical detection and biomedical diagnosis. Various sizing techniques have been widely used in sorting colloidal materials, analysing bioparticles and detecting the qualities of food and atmosphere. Most imaging-free methods for example light scattering measure the averaged size of particles and have difficulties in determining non-spherical particles. Imaging acquisition using camera is capable to observe individual particle in real time, but the accuracy is compromised by the image defocusing and instrumental calibration. In this Chapter, a machine learning-based pipeline is developed to facilitate a high accuracy imaging-based particle sizing. The pipeline consists of an image segmentation module for bioparticle identification and a machine learning model for accurate pixel-to-size conversion. The results manifest a significantly improved accuracy, showing great potential in a wide range of applications in environmental monitoring, biomedical diagnosis, and material characterization, etc.

5.1 Introduction

High accuracy size measurement is important for characterizing nanoscale and microscale particles. Various sizing measurement techniques are widely used to sort colloidal materials [178], analyse natural bioparticles such as pollen [179], characterize cells [47, 133, 180, 181], examine soil particle [182], monitor food quality during harvest [183], and assess air quality [184]. For instance, a golden standard for monitoring parasites in drinking water systems identifies different types of bioparticles characterized by their sizes [133]. Commonly used particle sizing techniques include non-imaging-based sizing techniques such as sieve analysis [185], static laser light scattering [186], dynamic light scattering [187], nanoparticle tracking analysis [188], time-of-transition (TOT) principle [189], as well as imaging-based sizing techniques such as bright-field microscopy [190], fluorescent microscopy [191, 192], and electron microscopy [193].

Sieve analysis [185, 194] is a traditional method used to measure the particle size. It utilizes stacked sieves with increasing aperture sizes to clamp particles and generate a size distribution. As shown in **Figure 5-1**, sieve analysis device is made of four sieves and each sieve has a specific open size. The measured objects are transported by gravity force with vibration, air entrainment or flowing liquid. Each sieve classifies a partial object into bin. When there are n sieves, the material can be divided into $n + 1$ particle size bins, and the particle size of each particle size bin is expressed by the corresponding sieve hole size of two adjacent sieves.

Other non-imaging-based sizing techniques estimate the particle size indirectly. For instance, static laser scattering [186] measures the gyration size instead of physical one based on the scattering pattern. Dynamic light scattering [187] retrieves the particle size based on the correlation function of scattered light signal,

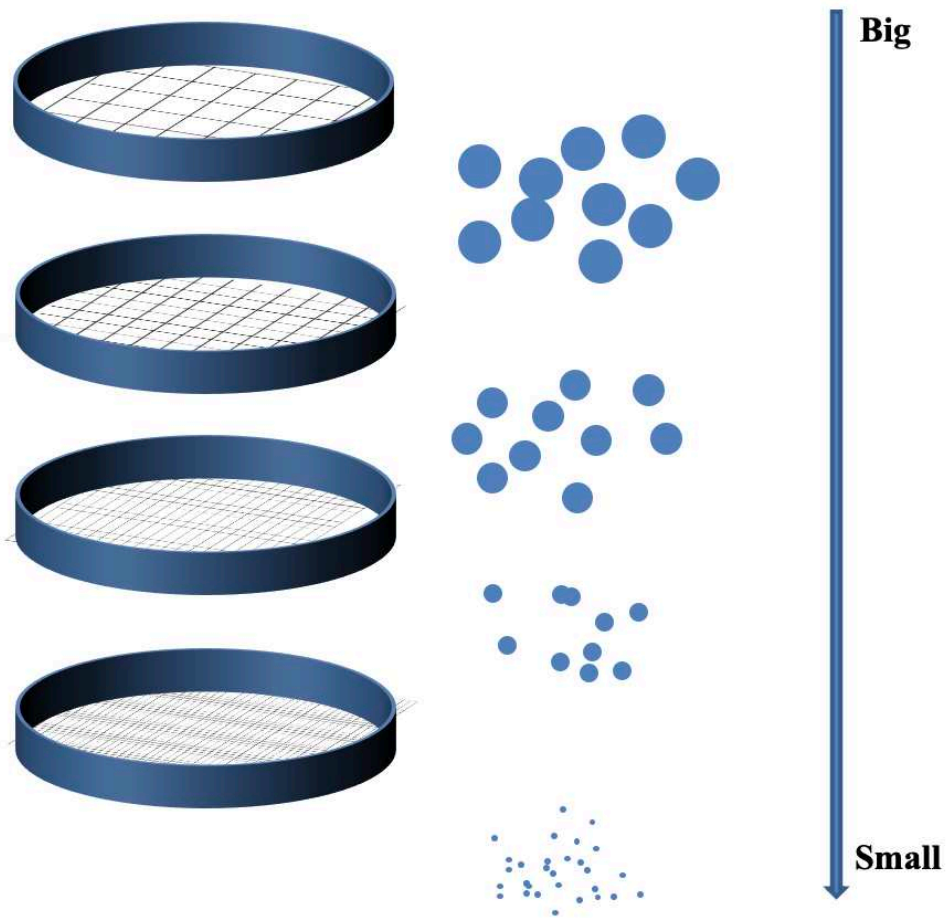


Figure 5-1: Schematic of sieve analysis device that is divided of four sieves, which each sieve reduces open size to classifier the objects into bins.

which essentially measures the diffusion coefficient of the bioparticle.

For nanoparticle tracking analysis [188], it is based on Brownian motion to obtain the size from the diffusion coefficient of particles by Stokes-Einstein equation. When light irradiates particles smaller than the wavelength, the light scatters in all directions around it. Since laser light is monochromatic and phase coherent, time-dependent fluctuations in scattering intensity can be observed from laser light. The fluctuation of the scattering intensity is mainly due to the changes in the relative position of the particles in the solution caused by Brownian motion.

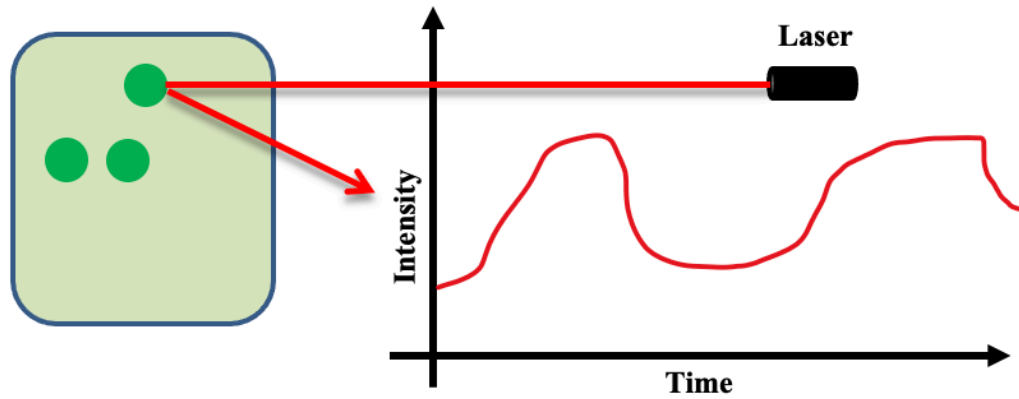


Figure 5-2: Large particles in dynamic light scattering. Time-dependent fluctuations in scattering intensity can be used to calculate size. Large particles move slow, so that the intensity of scattered light fluctuates slowly.

The Stokes-Einstein equation definition is given by

$$d = \frac{K_b T}{3\pi\eta_0 D} \quad (5-1)$$

where d is the hydrodynamic diameter, K_b is Boltzmann constant, T is the absolute temperature, η_0 is the viscosity of the system, and D is the diffusion coefficient. Large particles move slowly, and the intensity of scattered light fluctuates slowly as shown in **Figure 5-2**, while small particles move faster, and the intensity of scattered light fluctuates quickly as shown in **Figure 5-3**.

The non-imaging-based sizing techniques mentioned above are unable to accurately determine the size of non-spherical particles due to the limit of applied models. Conventional flow cytometry [17] determines the size according to the scattering pattern or other optical signatures such as time-of-transition. Before that, the calibration using particles of known size is required. Unfortunately, the calibration is not generalizable because particles of the same size may have substantially different optical signatures due to the difference in materials, surface properties, internal structures, or fluorescent labels.

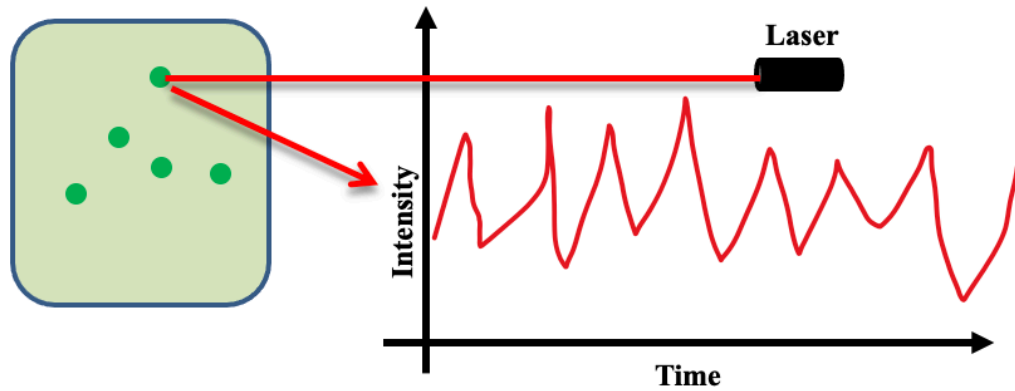


Figure 5-3: Small particle in dynamic light scattering. Time-dependent fluctuations in scattering intensity can be used to calculate size. Small particles move faster, so that the intensity of scattered light fluctuates more quickly.

Imaging-based sizing techniques such as imaging flow cytometry as shown in **Figure 5-4**, which is able to provide a direct measurement of the physical size of particles based on image analysis.

As illustrated in Figure 5-4, the optical system of an imaging flow cytometry includes a bright-field channel, a fluorescent channel and a side scatter channel. Bioparticles were hydrodynamically focused by a sheath flow and flowed through the detection region. Single bioparticles were illuminated with a LED light source and laser, and bright-field, fluorescent image, and side scatter image were acquired with a CCD/CMOS camera [142] using a $60\times$ objective. Each channel recodes 2D images with high speed simultaneously as shown in **Figure 5-5**.

Image analysis is operated automatically using intelligent software to analyse results of a huge number of images. Images of microscale and nanoscale particles are usually acquired using imaging-based microscopy. Commonly used image sensors include single-point photodetectors such as PMT and APD [195] as well as 2D photosensor arrays such as CCD or CMOS [196] as shown in **Figure 5-6**.

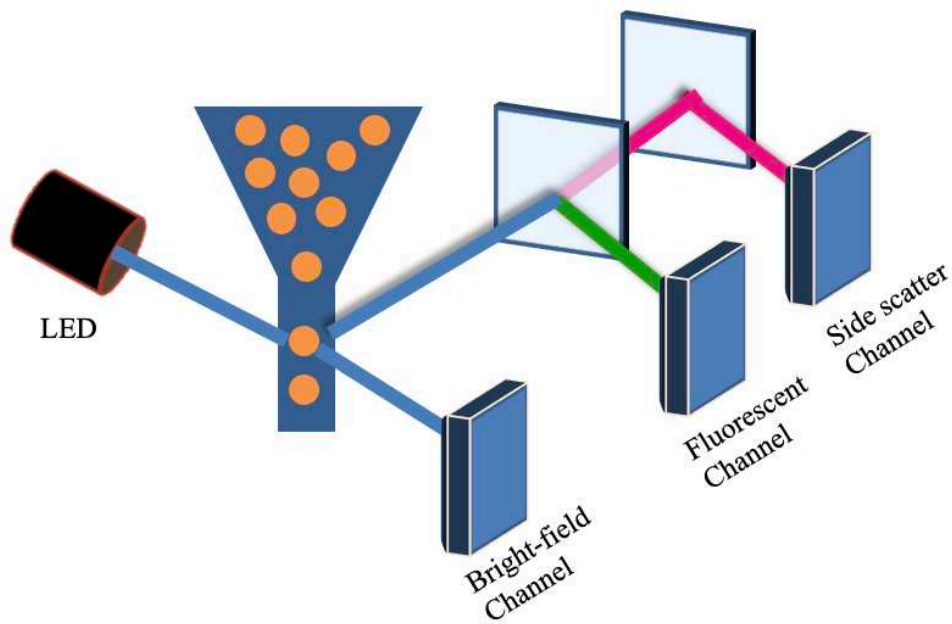


Figure 5-4: The schematic of the optical system for an imaging flow cytometry. The bioparticles pass through the imaging centre and each channel records 2D images from a particularly CCD or CMOS camera simultaneously.

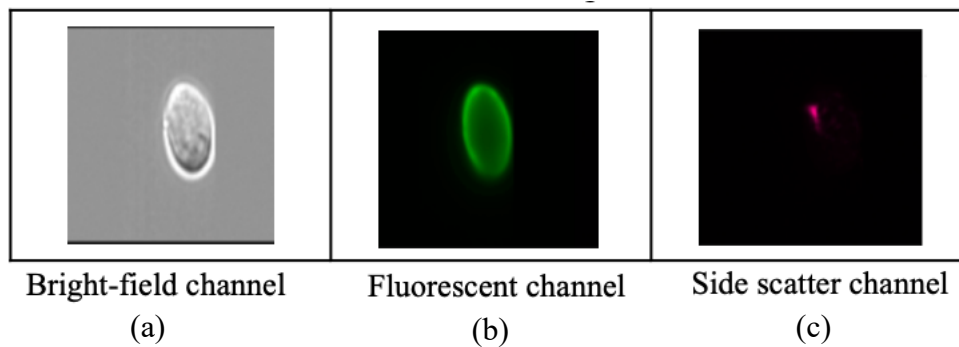


Figure 5-5: 2D images of different channel. Each channel records 2D image from particularly imaging modality. (a) bright-field image, (b) fluorescent image, (c) side scatter image.

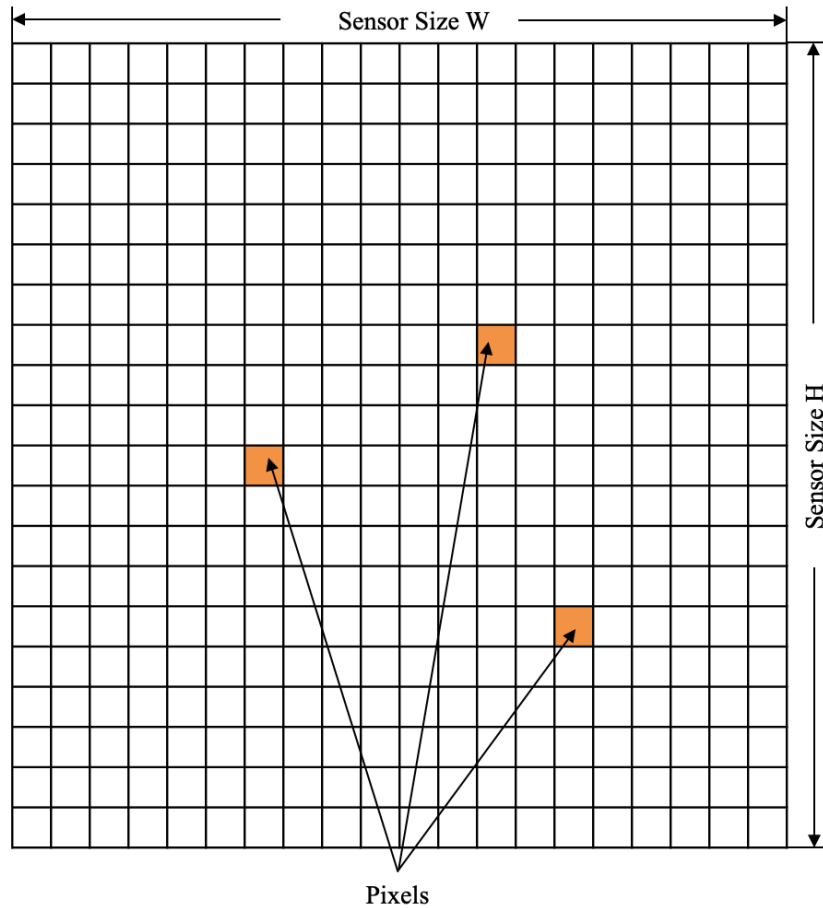


Figure 5-6: 2D CCD or CMOS camera. The 2D CCD/CMOS camera has a height H and a width W . Each pixel is a rectangle with a pixel size (PH and PW). Normally, PH equal to PW .

In a 2D sensing case (CCD or CMOS Sensor), the size of individual particle is estimated by converting the pixel to size at a fixed conversion ratio which is determined theoretically according to the specifications of the optical components. For example, a single pixel in images taken corresponds to $0.33 \mu\text{m}$ with a $60\times$ objective, and $0.5 \mu\text{m}$ with a $40\times$ objective according to the product specifications. However, it is noted that this fixed conversion ratio does not always give rise to an accurate particle sizing, probably arising from factors such as the objective error, imaging error, and segmentation error. Hence, the relationship between the pixel number and physical size is difficult to be modelled due to possible nonlinearity.

In this Chapter, the machine learning-based pipeline for imaging-based high accuracy particle sizing is studied. The machine learning-based pipeline automatically segments microparticles from the images, estimates the pixel size of particles, and predicts the physical size from the pixel information using a machine learning model to train there labelled images of calibration spherical beads. Compared to conventional approaches, our intelligent pipeline offers a more accurate particle sizing by learning from the massive calibration data. This machine learning-enabled pipeline would greatly extend the applicability of imaging-based sizing in the field of biomedical diagnostical, environmental sensing, and material characterization.

5.2 The Model of High Accuracy Bioparticle Sizing

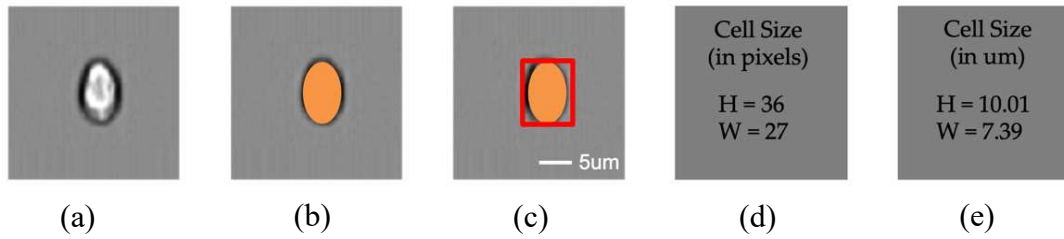


Figure 5-7: Different stage of size measurement pipeline. (a) Bioparticle image. (b) Segmentation. (c) Bioparticle Shape. (d) Bioparticle size in pixel numbers. (e) Bioparticle size in *um*. (a), (b) and (c) share the same scale bar.

The pipeline algorithm automatically analyzes the pixel information of the target particles and converts the pixel information into actual size based on a machine learning model is shown in **Figures 5-7**. First, it generates a contour of the particle using a segmentation algorithm. Then, the contour information is used to estimate the shape of the particle. Finally, the shape information is converted to physical length and width using the pixel-to-size module learnt by a quadratic machine learning model trained with least-squares regression [197] using the spherical beads of known sizes. All the aforementioned operations are integrated into an image processing pipeline to automatically predict the physical size of bioparticles from images acquired using an imaging flow cytometry (Amnis[®] ImageStream[®]X Mk II [43, 96]).

5.2.1 Segmentation and Pixel Measurement

Deep learning has recently made an impressive progress in imaging segmentation. For examples, U-Net [1], Deep Cell [198], Faster R-CNN [199], Mask R-CNN [111], and RetinaNet [200] have been demonstrated for instance segmentation in single bioparticle analysis [201]. However, those deep learning models are computationally intensive and require heavy labelling from human. Imaging flow cytometry is capable of generating single cell image with a clear background. Hence

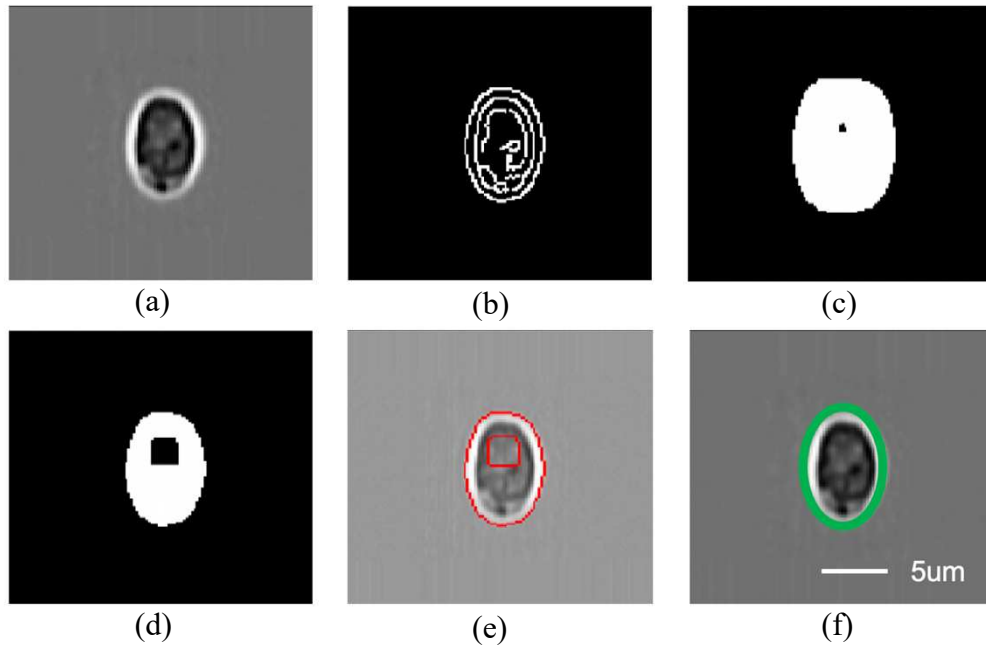


Figure 5-8: Different processing stage of segmentation and pixel measurement: (a) Gaussian blur, (b) canny detector, (c) erode, (d) dilate, (e) find contours, and (f) estimate shape. All subfigures are in the same scale bar.

it is well-suited for computer-vision-based analysis.

The computer-vision-based segmentation algorithm (**Figure 5-8**) first resizes the input single-particle images into 120×120 pixels and removes the noise using a Gaussian blurring module (**Figure 5-8a**). Then, a Canny detector is applied to the processed images to generate the edge images (**Figure 5-8b**) that are subsequently processed with erode (**Figure 5-8c**) and dilating (**Figure 5-8d**) algorithms to generate the output blob images. Next, the algorithm identifies the edge in the blob images and generates the contour information of the particle (**Figure 5-8e**). The height and width in terms of pixel numbers are estimated from particle contour (**Figure 5-8f**). In the case of spherical particles, the height and width have the same value. Finally, the physical size of the particles is determined based on the machine learning model.

Signal noise often degrades the image quality and introduces error to subsequent processing submodules in the pipeline. Gaussian blur [202] is a popular

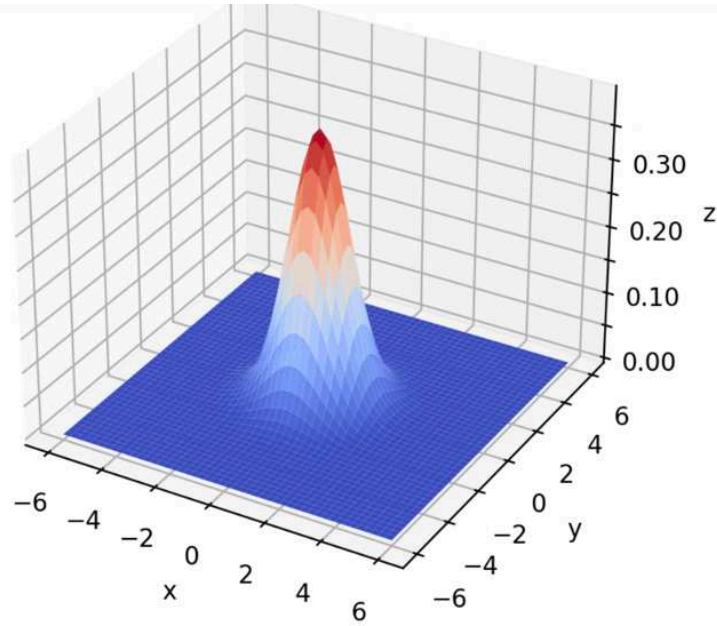


Figure 5-9: A Gaussian kernel with $\sigma = 0.25$.

algorithm to reduce the noise and enhance the image quality. The formula of the Gaussian blur is expressed as [202]

$$g(x, y) = \sum_{x,y} f(x - i, y - j)h(i, j) \quad (5-2)$$

where $g(x, y)$ is the output pixel value, $f(x, y)$ is the input image pixel, and $h(i, j)$ is a Gaussian kernel given by [202]

$$h(i, j) = A \exp\left(-\left(\frac{(i-i_0)^2}{2\sigma_i^2} + \frac{(j-j_0)^2}{2\sigma_j^2}\right)\right) \quad (5-3)$$

where A is the amplitude of the gaussian kernel, i_0 and j_0 mark the center position of the kernel, and σ represents the standard deviation (SD) with respect to variables i and j . An example of Gaussian kernel with $\sigma = 0.25$ is shown in **Figure 5-9**.

Canny detector [203] is a popular technique in edge detection given its advantages in low error rate, high localizability, and minimized response. The Canny edge detection algorithm contains gradient calculation, rounded and a non-maximum suppression. The gradient strength G and direction θ are expressed as [203]

$$G = \sqrt{G_x^2 + G_y^2} \quad \text{and} \quad \theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (5-4)$$

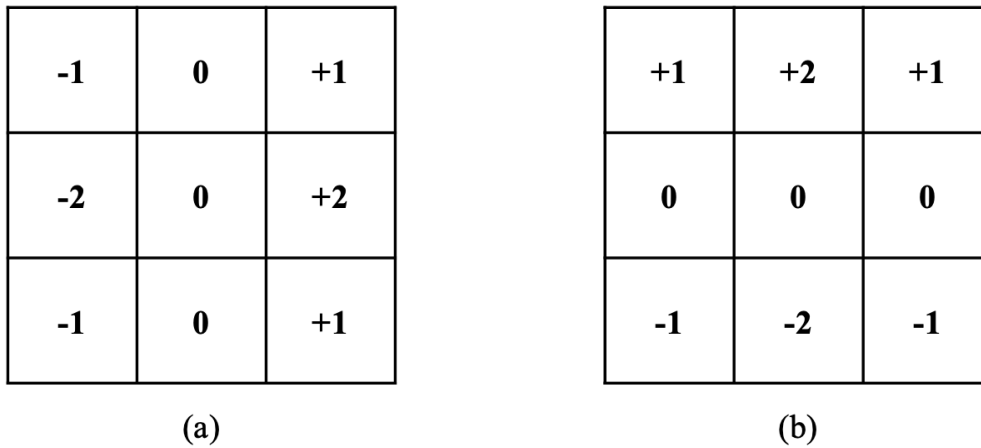


Figure 5-10: Horizontal and vertical direction kernels. (a) G_x and (b) G_y .

where the G_y and G_x are the first derivatives of vertical direction (G_y) and horizontal direction (G_x), respectively. The gradient kernels used for calculating the gradient G_x and G_y are shown in **Figure 5-10**.

The θ is rounded to 0, 45, 90 or 135 degrees. For example, the θ in between 22.5 degree to 67.5-degree are mapped to 45-degree. Next, a non-maximum suppression algorithm is applied to remove non-considered pixel so that only the thin lines remain. Finally, a hysteresis stage with high and low threshold is applied on the lines to further improve the results.

Dilate and Erode [204] are two basic morphological operations for removing noise, isolating or jointing the individual components, and finding the intensity bumps or holes in an image. The dilate operation uses a kernel, such as 3×3 pixels, with an anchor point at the center of the kernel to scan over the image and calculate the maximum pixel value. That maximum value replaces the value in the anchor point. As a result, the bright regions are expanded, and the individual components with small gaps in between are connected. In contrast, the erode operation uses the minimal value to replace the value in the anchor point to render a thinner bright area.

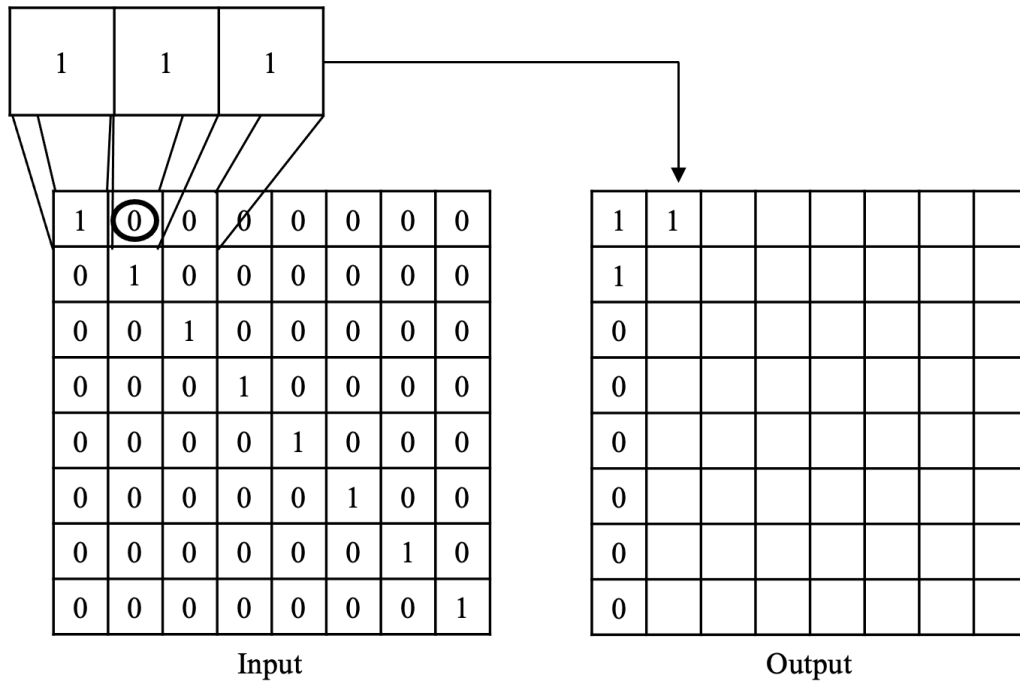


Figure 5-11: Dilate operation.

The example of dilate is shown in **Figure 5-11**. The black circle is the current pixel. The output is the maximum number of the neighboring pixels. And the example of erode is shown in **Figure 5-12**. The black circle is the current pixel. The output is the minimum number of the neighboring pixels.

The find contours operation [205] obtains the contour information. A contour is a closed curve where all its points are on the boundary and have the same value. In our algorithm, ellipse is used to approximate the outline of the cells. In the last stage of the imaging processing, the contour information of the cells is passed into an estimator function to obtain the inscribed rotated rectangle of the ellipse.

5.2.2 Conversion of Pixel Size to Physical Size

The size converter algorithm converts the pixel to size in micrometers with the machine learning model. The calibration process started with the collecting images of microplastic beads with diameters of 3 μm , 4 μm , 4.6 μm , 5 μm , 5.64 μm ,

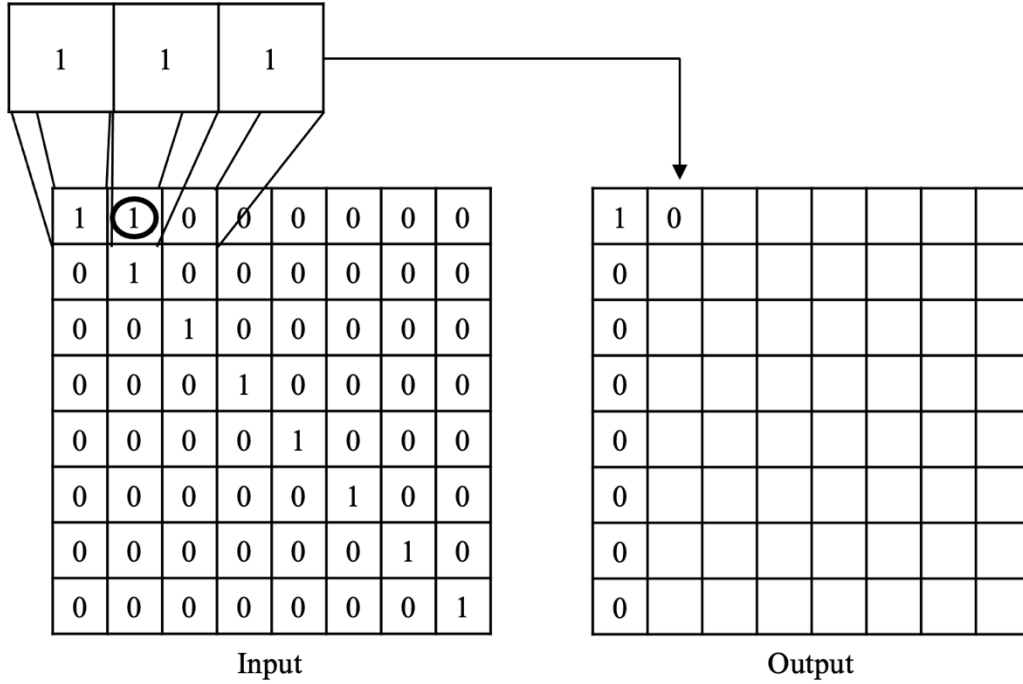


Figure 5-12: Erode operation.

7.32 μm , 8 μm , 10 μm , 12 μm , and 15 μm (from Thermo Fisher Scientific, Duke Scientific and Polysciences Inc.). Then, they were processed with the segmentation algorithm to generate the beads diameters in pixels. Finally, parameters a , b , and c of the quadratic curve model $y = ax^2 + bx + c$ was learned by minimalizing the error which is expressed as

$$\begin{aligned}
 Error &= \varepsilon(a, b, c) \\
 &= \sum_{i=1}^n (ax_i^2 + bx_i + c - y_i)^2
 \end{aligned} \tag{5-5}$$

The partial derivative with a can be calculated by

$$\frac{\partial \varepsilon(a, b, c)}{\partial a} = \sum_{i=1}^n 2(ax_i^2 + bx_i + c - y_i)x_i^2 \tag{5-6a}$$

$$\begin{aligned}
 &= 2 \left[a \sum_{i=1}^n x_i^4 + b \sum_{i=1}^n x_i^3 + c \sum_{i=1}^n x_i^2 \right. \\
 &\quad \left. - \sum_{i=1}^n y_i x_i^2 \right] \\
 &= 0
 \end{aligned}$$

From the partial derivative and rearranging gives

$$\left[\sum_{i=1}^n x_i^2 \right] c + \left[\sum_{i=1}^n x_i^3 \right] b + \left[\sum_{i=1}^n x_i^4 \right] a = \sum_{i=1}^n y_i x_i^2 \quad (5-6b)$$

The partial derivative with b can be calculated by

$$\begin{aligned}
 \frac{\partial \varepsilon(a, b, c)}{\partial a} &= \sum_{i=1}^n 2(ax_i^2 + bx_i + c - y_i)x_i \\
 &= 2 \left[a \sum_{i=1}^n x_i^3 + b \sum_{i=1}^n x_i^2 + c \sum_{i=1}^n x_i - \sum_{i=1}^n y_i x_i \right] \\
 &= 0
 \end{aligned} \quad (5-7a)$$

Base on the partial derivative, the Equation 5-7a can be rearranged as

$$\left[\sum_{i=1}^n x_i \right] c + \left[\sum_{i=1}^n x_i^2 \right] b + \left[\sum_{i=1}^n x_i^3 \right] a = \sum_{i=1}^n y_i x_i \quad (5-7b)$$

The partial derivative deal with c can be calculated by

$$\begin{aligned}
 \frac{\partial \varepsilon(a, b, c)}{\partial a} &= \sum_{i=1}^n 2(ax_i^2 + bx_i + c - y_i) \\
 &= 2 \left[a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i + cn - \sum_{i=1}^n y_i \right] \\
 &= 0
 \end{aligned} \quad (5-8a)$$

From the partial derivative and rearranging gives

$$nc + \left[\sum_{i=1}^n x_i \right] b + \left[\sum_{i=1}^n x_i^2 \right] a = \sum_{i=1}^n y_i \quad (5-8b)$$

By use the linear algebra, the matrix equation regarding the least-square regression parameters of a, b, and c in quadratic curve $y = ax^2 + bx + c$ can be given by

$$\begin{bmatrix} n & s_1 & s_2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix} \begin{bmatrix} c \\ b \\ a \end{bmatrix} = \begin{bmatrix} z \\ \gamma_{yx} \\ \gamma_{yx^2} \end{bmatrix} \quad (5-9)$$

where $s_1, s_2, s_3, s_4, z, \gamma_{yx}$, and γ_{yx^2} can be expressed as

$$s_k = \sum_{i=1}^n x_i^k \quad (5-10a)$$

$$z = \sum_{i=1}^n y_i \quad (5-10b)$$

$$\gamma_{yx^j} = \sum_{i=1}^n y_i x_i^j \quad (5-10c)$$

where x_i is the pixels size of the individual bead and y_i is the corresponded physical size of the bead, and n is the total number of beads.

When the linear models are learned, we obtained the parameters $m = 0.2905$ and $b = 0.4785$ for the linear model and $a = -0.000163$, $b = 0.301$, and $c = -0.618$ for the quadratic curve model. As the RMSE of the quadratic model is smaller than the linear model (0.2657 vs 0.2668), the quadratic curve model was adopted to implement the size converter module.

5.3 Experimental Results and Discussions

5.3.1 Performance Evaluation

To evaluate the performance of the image processing pipeline, the image database of microplastic beads of known sizes and biological cells have been built. First, the image segmentation algorithm was evaluated with the Intersection over Union (IoU) metric [206]. The performance of the machine learning model was evaluated with Root Mean Square Error (RMSE) [207]. And the measurement on a realistic cell dataset was performed. The mathematical expressions of IoU and RMSE metrics are expressed as

$$IoU = \frac{target \cap prediction}{target \cup prediction} \quad (5-11)$$

and

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (5-12)$$

where *target* is the area in ground truth and *prediction* is the target segmented area; y_j is the physical size, \hat{y}_j is the predicted size and n is the total number of particles.

The graphic representation of IoU is shown in **Figure 5-13**. Furthermore, the height and width distributions of particles such as beads, *Cryptosporidium* and *Giardia* oocytes are determined using bright-field imaging flow cytometry.

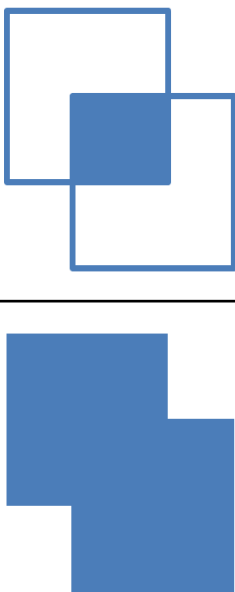
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (5-13)$$


Figure 5-13: Graphic representation of Intersection over Union (IoU).

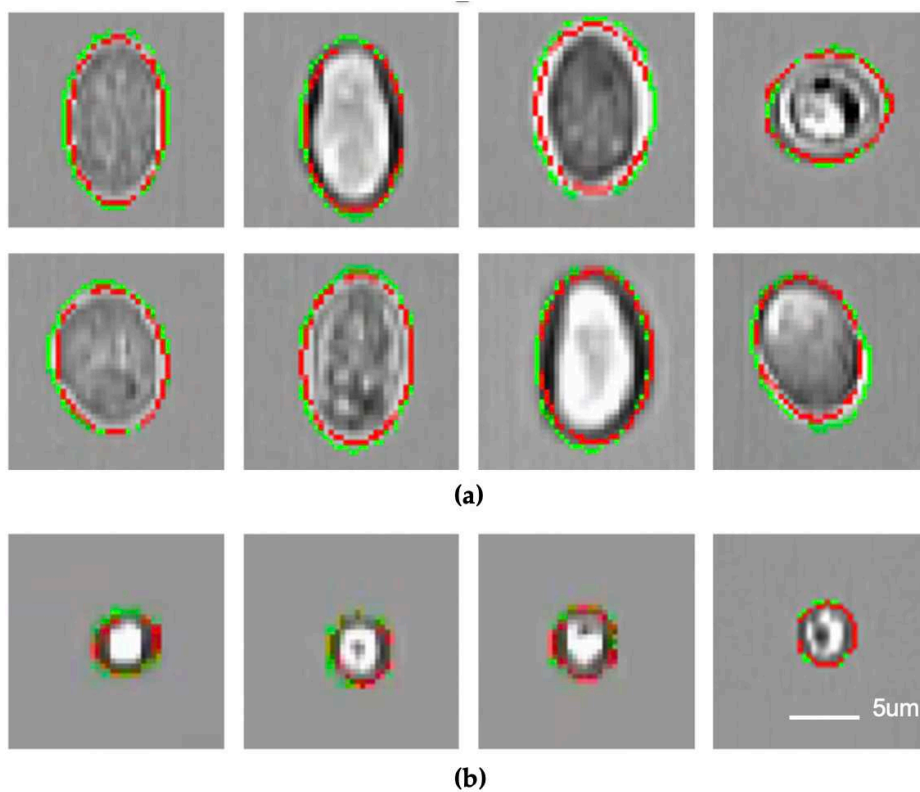


Figure 5-14: Error analysis of segmentation algorithm. The image output of the segmentation algorithm of: (a) Segmentation results of *Giardia*, and (b) Segmentation results of *Cryptosporidium*. All subfigures are in the same scale bar.

5.3.2 Segmentation and Pixel Size Measurement

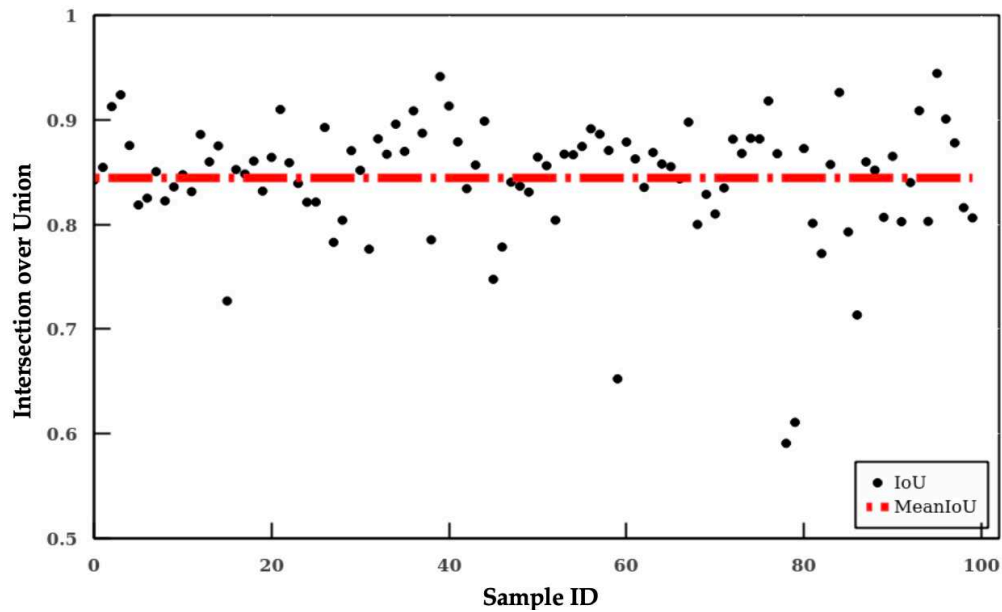


Figure 5-15: Intersection over Union results of the segmentation algorithm.

The segmentation results are evaluated with the IoU score between the contour labelled by human operators and the contour predicted by the algorithm. The output of the segmentation algorithm is depicted in **Figure 5-14**. The top two rows are the results of segmented *Giardia oocyte* images, and the lowest row is the results of segmented *Cryptosporidium oocyte* images. In these images, the green line is the ground truth (human labeled), and the red line is the output of the segmentation algorithm. As shown in **Figure 5-14**, the image outputs of the segmentation algorithm are close to the ground truth. Overall, the segmentation algorithm achieved 84.4% in mean IoU (red dotted line) as shown in **Figure 5-15**, in which each black dot represents the IoU of an individual image output of the testing dataset.

5.3.3 Physical Size Measurement

The imaging flow cytometer uses a fixed pixel-to-size ratio based on the specifications of the optics for particle sizing. However, this approach often leads to large errors in particle size (**Table 5-1**). Therefore, a machine learning model is

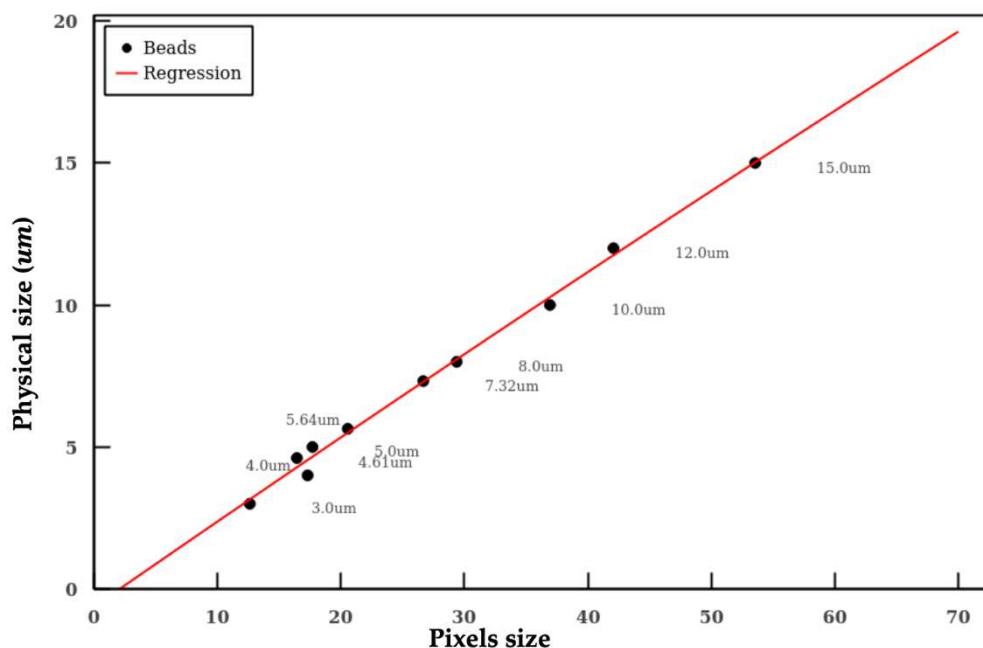


Figure 5-16: Quadratic curve-based calibration.

established to determine the pixel-to-size ratio for accurate sizing. Both linear and quadratic regression models are adopted to learn the relationship between the pixel (pixels) and length (μm) of microplastic beads of known sizes. As the RMSE of the quadratic model is smaller than the linear model (0.2657 vs 0.2668), the quadratic curve model was employed. **Figure 5-16** shows the diameter versus the pixel size of the microplastic beads. The quadratic machine learning regression model is shown as the red curve.

The sizes of the microplastic beads measured using our algorithm and using the fixed pixel-to-size ratio ($0.33 \mu m/\text{pixel}$ with $60\times$ objective on Amnis Imagestream MKII) are summarized in **Table 5-1** and **Figure 5-17**. The fixed pixel-to-size conversion ratio is the mainstream approach used by imaging flow cytometry. Our algorithm shows significantly more accurate sizing in comparison. **Figure 5-17** shows the length distribution in both axes of microplastic particles within a distribution range. The red dots represent the actual sizes of beads (ground truth based

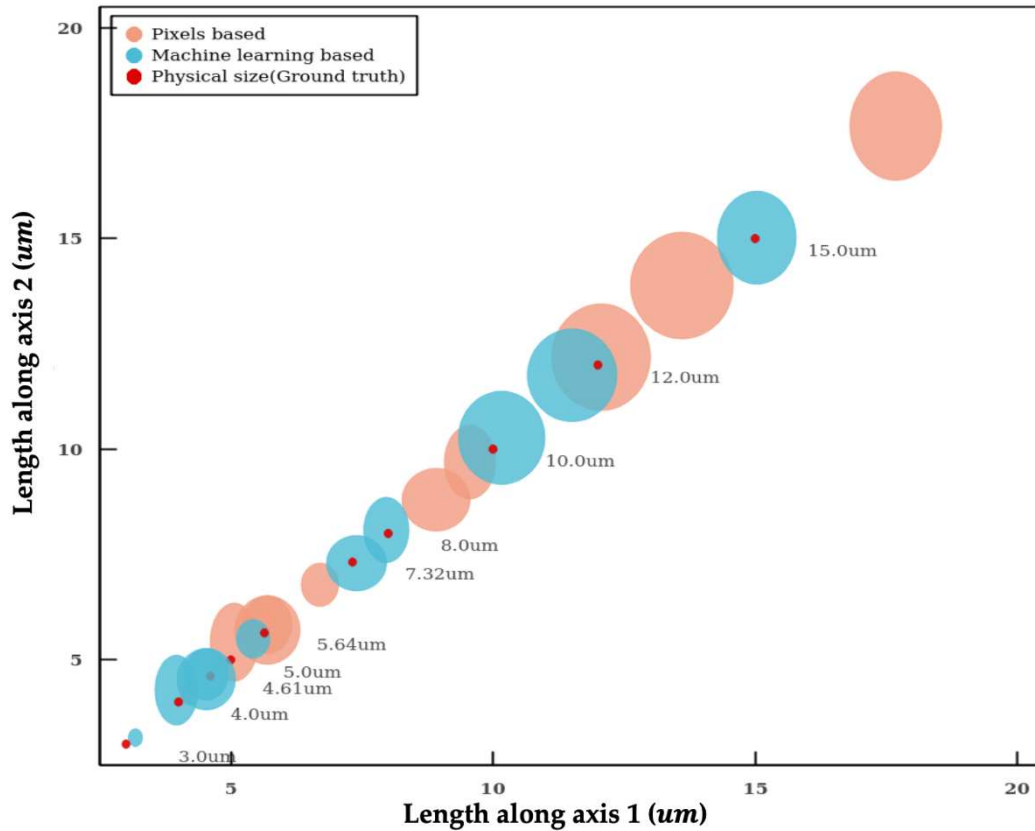


Figure 5-17: The length distribution in both axes of microplastic particles within a distribution range. The circle represents the population distribution with the error in 2σ range. Red color ones are the physical diameters of beads, pink color ones are the beads sizes based on assumption, and cyan color ones are the beads sizes based on machine learning calibration.

on manufacturer’s specifications), the brown dots present the size measured using the fixed pixel-to-size ratio, and the dark green dots represent the size measured using the machine learning model. The sizes of the dots represent the SD of the measurement. The microplastic particles have a narrow distribution with a CV < 2% according to product specifications.

The machine learning model gives rise to significantly more accurate size measurement compared to the approach using fixed pixel-to-size ratio. The sizes of microplastic beads measured using the machine learning model deviate only slightly from the ground truth with a mean percentage error of 4.2% (Table 5-1). In contrast,

the mean percentage error using the fixed conversion ratio is 23.3% which is 5 times larger than the machine learning model.

As shown in **Figure 5-17**, methods using fixed conversion ratio tends to overestimate the size of the particle. In the worst scenario, the percentage error even reaches a value close to 40%. In addition, the SD measured with the machine learning model is also smaller in comparison, which indicates a better precision of particle sizing. The individual measurements of microplastic beads of $3\ \mu\text{m}$, $5\ \mu\text{m}$, $12\ \mu\text{m}$ and $15\ \mu\text{m}$ using the machine learning model are shown in **Figure 5-18**.

Table 5-1: Measurement error analysis

Beads	Learnt model			Fixed ratio ($0.33\ \mu\text{m}/\text{pixel}$)		
	μ	σ	Error (%)	μ	σ	Error (%)
$3.0\ \mu\text{m}$	3.15	0.21	5.1	4.16	0.24	38.8
$4.0\ \mu\text{m}$	4.54	0.73	13.5	5.71	0.82	42.7
$4.6\ \mu\text{m}$	4.28	0.83	7.1	5.42	0.93	17.6
$5.0\ \mu\text{m}$	4.66	0.61	6.9	5.84	0.69	16.8
$5.64\ \mu\text{m}$	5.50	0.46	2.5	6.78	0.51	20.2
$7.32\ \mu\text{m}$	7.29	0.66	0.4	8.80	0.74	20.2
$8.0\ \mu\text{m}$	8.08	0.77	1.0	9.69	0.88	21.2
$10.0\ \mu\text{m}$	10.27	1.11	2.7	12.18	1.27	21.8
$12.0\ \mu\text{m}$	11.75	1.10	2.1	13.89	1.27	15.7
$15.0\ \mu\text{m}$	15.02	1.10	0.1	17.67	1.29	17.8
Avg.		0.76	4.2		0.86	23.3

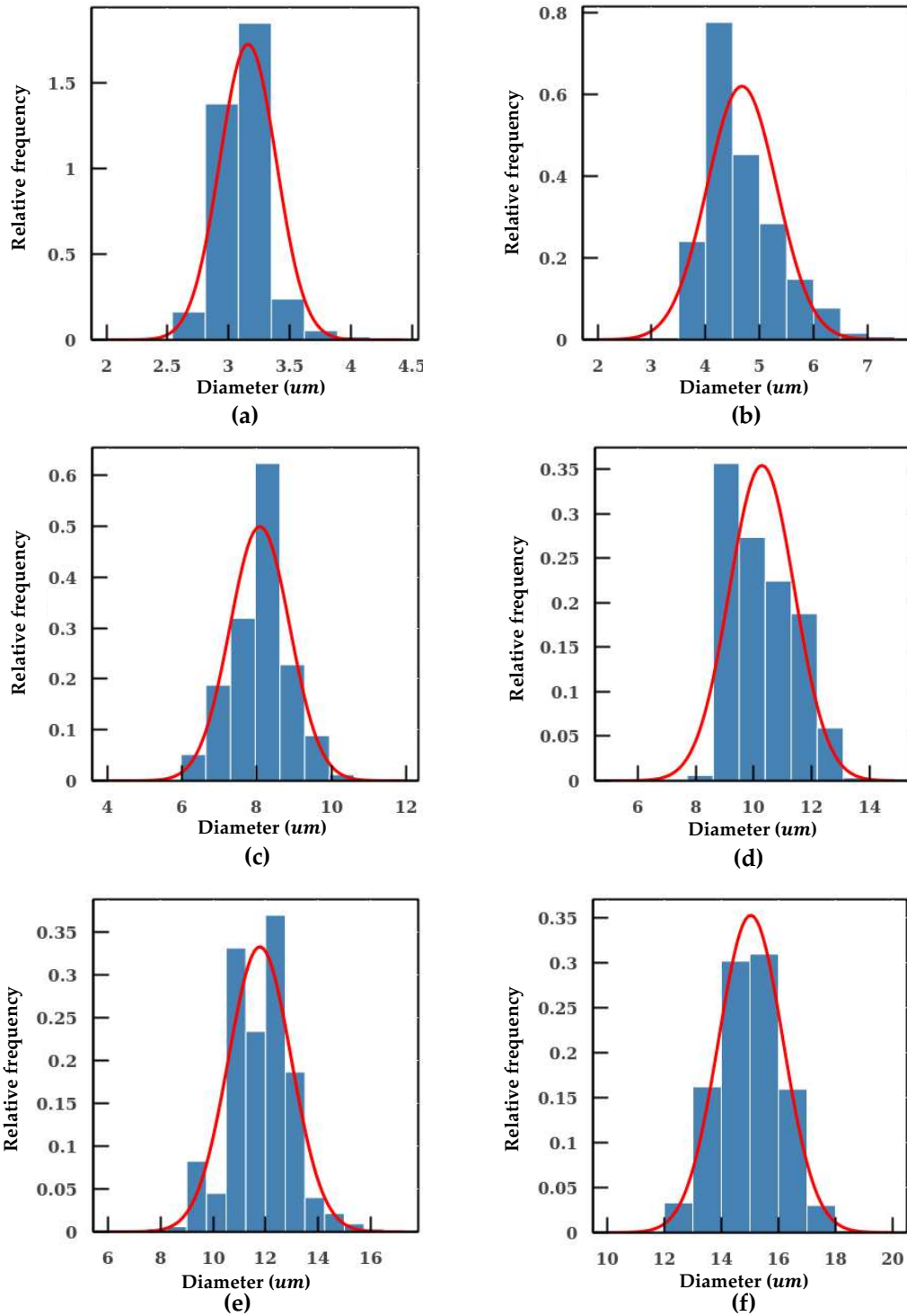


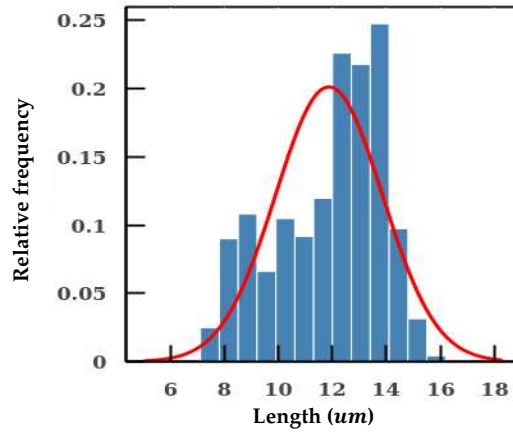
Figure 5-18: Measurement of individual microplastic particles sizes distribution: (a) 3 *um*, (b) 5 *um*, (c) 8 *um*, (d)10 *um*, (e)12 *um*, and (f)15 *um*.

The aforementioned algorithms are integrated into a pipeline. With this intelligent pipeline, the height and width distributions of *Cryptosporidium* and

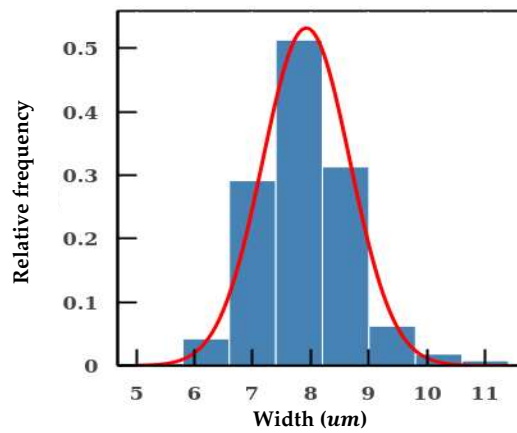
Giardia oocytes are determined using bright-field images from the imaging flow cytometry. The results are presented in **Figure 5-17** and **Table 5-2**. Our intelligent pipeline determines that the mean height of *Giardia oocytes* is $11.87 \mu\text{m}$ with a SD of $1.9 \mu\text{m}$. The mean width of the *Giardia oocytes* is $7.92 \mu\text{m}$ with a SD of $0.75 \mu\text{m}$. The *Cryptosporidium oocytes* are approximately spherical, and the mean diameter *Cryptosporidium oocytes* measured using our algorithm is $5.03 \mu\text{m}$ with a SD of $0.48 \mu\text{m}$. In contrast, the mean height and width of *Giardia oocytes* are $12.94 \mu\text{m}$ and $8.45 \mu\text{m}$, and the mean diameter *Cryptosporidium oocytes* is $5.17 \mu\text{m}$ when calculated using the fixed conversion ratio.

Table 5-2: Measurement results on bioparticles

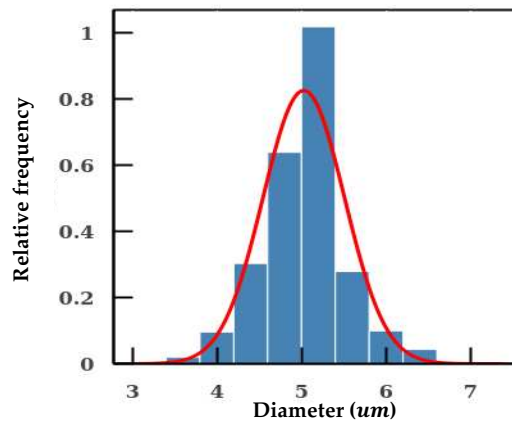
	Feature	Mean (μm)	SD (μm)
<i>Giardia</i>	Height	11.87	1.99
	Width	7.92	0.75
<i>Cryptosporidium</i>	Diameter	5.03	0.48



(a)



(b)



(c)

Figure 5-19: Measurement on *Cryptosporidium* and *Giardia*. (a) Length of the *Giardia*. (b) Width of the *Giardia*. (c) Diameter of the *Cryptosporidium*.

5.4 Summary

In this Chapter, a machine learning-based pipeline for imaging-based high-accuracy bioparticle sizing is demonstrated. The pipeline consists of an image segmentation module for extracting contours and estimating the pixel size of the bioparticle as well as a machine learning model for accurate pixel-to-size conversion. The computer vision-based segmentation algorithm first resizes the input single-particle images into 120×120 pixels and removes the noise using a Gaussian blurring. Then, a Canny detector is applied to the processed images to generate the edge images that are subsequently processed with erode and dilating algorithms to generate the output blob images. The segmentation results are evaluated with the IoU score between the contour labelled by human operators and the contour predicted by the algorithm. The image segmentation algorithm achieves 84.4% in the mean IoU. The size converter module converts the pixel to size in micrometers with the machine learning model trained by the collected images of calibration beads. The performance of the machine learning model was evaluated with Root Mean Square Error (RMSE). Final, the particle size determined by the machine learning model only has a mean percentage error of 4.2% which is five times better than the method using a fixed pixel-to-size conversion ratio (23.3%). Our method empowers different intelligent imaging systems such as imaging flow cytometry for high-accurate particle sizing and promises great potential in a wide range of applications in the field of environmental sensing, biomedical diagnostics, and material characterization.

Chapter 6 Concluding Remarks

6.1 Conclusions

In this PhD project, three areas in machine learning have been studied and demonstrated to detect or measure the size of the bioparticles for various applications including (1) Deep learning-enable high-speed bioparticle detection; (2) deep metric learning for rare bioparticle detection; (3) machine learning-based pipeline for bioparticle sizing. Model theoretical, design of the model, experimental results are discussed.

In the study of deep learning-enable high-speed bioparticle detection, image classification models of the deep learning are derived. a deep learning-enabled high-throughput system are developed for predicting *Cryptosporidium* and *Giardia* in drinking water. This system combines imaging flow cytometry and an efficient artificial neural network called MCellNet, which achieves a classification accuracy > 99.6%. The system can detect *Cryptosporidium* and *Giardia* with a sensitivity of 97.37% and a specificity of 99.95%. The high-speed analysis reaches 346 frames per second, outperforming the state-of-the-art deep learning algorithm MobileNetV2 in speed (251 frames per second) with a comparable classification accuracy.

In the study of deep metric learning for rare bioparticle detection, a model of deep metric learning for rare bioparticle detection is derived and the deep metric learning algorithm leverages convolutional neural network to study the rich feature inside the dataset and learns distinct metric by using Siamese network and contrastive loss which learned to maximum the distance of different classes and minimal the

distance of similar class. Deep metric learning offers a generatability and uses distance information to model the similarity of the images by learning a function maps from images pixels to latent space, playing a vital role in rare object detection. Experiments show that the deep metric neural network achieved a high accuracy of 99.86% in classification, 98.89% in precision rate, 99.16% in recall rate and zero false alarm rate for rare bioparticle detection.

In the study of machine learning-based pipeline for bioparticle sizing, an intelligent pipelined for bioparticle sizing by employing computer vision and machine learning is built up and the intelligent pipeline automatically identifies and segments particles from the micrographs, measures the pixel size of the particles, and converts the pixel into actual size using a machine learning model learned from mass training data. Compared to the conventional approaches, the particle size determined by the machine learning model only has a mean percentage error of 4.2% which is five times better than the methods using a fixed pixel-to-size conversion ratio (23.3%). The method empowers different intelligent imaging systems such as imaging flow cytometry for high accurate particle sizing and promises great potential in a wide range of applications in the field of environmental sensing, biomedical diagnostics, and material characterization.

Moreover, a bright-field image database of *Cryptosporidium parvum* oocysts (2,082 images) and *Giardia lamblia* cyst (3,569 images), 1.54-um beads (3,466 images), 3-um beads (3,457 images), 4-um beads (5,783 images), 4.6-um beads (2,188 images), 5-um beads (9,637 images), 5.64-um beads (3,285 images), 8-um beads (3,066 images), 10-um (8,270 images), 12-um (4,704 images), 15-um beads (2,813 images), and natural pollutants of various shape and sizes (27,826 images) are built. Each image is patched to a standard size with 120×120 pixels and it can be

used to test bioparticle detection algorithms.

These works empower rapid, accurate, and high throughput bioparticle detection and sizing, they promise great potential in a wide range of applications in the field of biomedical diagnostics, environmental monitoring, and material characterization, etc.

6.2 Recommendations

Recommendations for future work are summarized as follows:

6.2.1 Efficient Neural Networks Design

Efficient neural networks (DNNs) [208, 209] become critical for embedded applications due to their low storage requirement and computing efficiency. In recent years, the performance of deep neural networks have been greatly improved, and the storage and computing complexity of the network have also increased. How to accelerate and compress deep neural networks and improve the operating efficiency of deep neural networks have become a research hotspot in the field of deep learning. In response to this problem, with the joint efforts of academia and industry, a series of methods have been proposed. Besides the compact network design, other approaches, such as network pruning [210, 211], low-rank decomposition [212-215], network quantification [216-218], knowledge transfer network [219, 220], are also valuable to be explored.

6.2.2 Network Improving

Supervised machine learning, especially deep neural network, is domestic today's applications in solving the real problem [221, 222]. Unfortunately, in the traditional machine learning, there exist two main uncertainties: *aleatoric* uncertainty and *epistemic* uncertainty. The *aleatoric* uncertainty is the uncertainty inherent in the data if there are some randomness. The *epistemic* uncertainty is due to lack of data and cannot estimate the model parameter without any doubt. The model may fail in novel data. Deep learning with metric learning is an important approach for those issues. Bayesian deep neural network is another approach [223]. Bayesian deep neural network can be understood as regularization by introducing uncertainty into

the weight of the neural network, which is equivalent to ensemble of infinite groups of neural networks on a certain weight distribution for prediction. For example, in the conventional deep neural network, all the parameters are a value, and the predicted output of the network is also a value. In contrast, all parameters of the Bayesian deep neural network are a distribution, and the predicted output of the network is also a distribution. An conventional deep neural network is to fit the data sample with maximum likelihood estimation (MLE).

The difference between Bayesian deep neural network and conventional deep neural network is that its weight parameter is a random variable, rather than a definite value. It is a combination of probabilistic modelling and neural network, and can give the confidence of the prediction result. A priori is used to describe the key parameters and the input of the neural network. Bayesian deep neural network is very critical for many problems, because BNN has the ability to quantify uncertainty. It is very robust and it is a promised approach for further study in bioparticle detection.

Bibliography

1. Falk, T., et al., *U-Net: deep learning for cell counting, detection, and morphometry*. Nat Methods, 2019. **16**(1): p. 67-70.
2. LeCun, Y., et al., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
3. Heo, Y.J., et al., *Real-time image processing for microscopy-based label-free imaging flow cytometry in a microfluidic chip*. Scientific reports, 2017. **7**(1): p. 11651.
4. Isozaki, A., et al., *Intelligent image-activated cell sorting 2.0*. Lab on a Chip, 2020. **20**: p. 2263-2273.
5. Zhang, Y., et al., *Computational cytometer based on magnetically modulated coherent imaging and deep learning*. Light: Science & Applications, 2019. **8**(1): p. 1-15.
6. Viola, P. and M. Jones, *Rapid object detection using a boosted cascade of simple features*. CVPR (1), 2001. **1**(511-518): p. 3.
7. Felzenszwalb, P., D. McAllester, and D. Ramanan. *A discriminatively trained, multiscale, deformable part model*. in *2008 IEEE conference on computer vision and pattern recognition*. 2008. IEEE.
8. Jo, Y., et al., *Holographic deep learning for rapid optical screening of anthrax spores*. Sci Adv, 2017. **3**(8): p. e1700606.

9. van Dongen, J.J.M., et al., *Minimal residual disease diagnostics in acute lymphoblastic leukemia: need for sensitive, fast, and standardized technologies*. Blood, The Journal of the American Society of Hematology, 2015. **125**(26): p. 3996-4009.
10. Kanegane, H., et al., *Flow cytometry-based diagnosis of primary immunodeficiency diseases*. Allergology International, 2018. **67**(1): p. 43-54.
11. Jennings, C.D. and K.A. Foon, *Recent advances in flow cytometry: application to the diagnosis of hematologic malignancy*. Blood, The Journal of the American Society of Hematology, 1997. **90**(8): p. 2863-2892.
12. Seo, K.H., R.E. Brackett, and J.F. Frank, *Rapid detection of Escherichia coli O157: H7 using immuno-magnetic flow cytometry in ground beef, apple juice, and milk*. International Journal of Food Microbiology, 1998. **44**(1-2): p. 115-123.
13. Buzatu, D.A., et al., *An integrated flow cytometry-based system for real-time, high sensitivity bacterial detection and identification*. PloS one, 2014. **9**(4): p. e94254.
14. Gillespie, S., et al., *Assessing microbiological water quality in drinking water distribution systems with disinfectant residual using flow cytometry*. Water Research, 2014. **65**: p. 224-234.
15. Helmi, K., et al., *Assessment of flow cytometry for microbial water quality monitoring in cooling tower water and oxidizing biocide treatment efficiency*. Journal of microbiological methods, 2018. **152**: p. 201-209.

16. Safford, H.R. and H.N. Bischel, *Flow cytometry applications in water treatment, distribution, and reuse: A review*. Water research, 2019. **151**: p. 110-133.
17. Adan, A., et al., *Flow cytometry: basic principles and applications*. Critical Reviews in Biotechnology, 2017. **37**(2): p. 163-176.
18. Zucker, R., et al., *Detection of TiO₂ nanoparticles in cells by flow cytometry*. Cytometry Part A, 2010. **77**(7): p. 677-685.
19. Shapiro, H.M., *Practical flow cytometry*. 2005: John Wiley & Sons.
20. McKinnon, K.M., *Flow cytometry: an overview*. Current protocols in immunology, 2018. **120**(1): p. 5.1. 1-5.1. 11.
21. Macey, M.G. and M.G. Macey, *Flow cytometry*. 2007: Springer.
22. Marrison, J., et al., *Ptychography—a label free, high-contrast imaging technique for live cells using quantitative phase information*. Scientific Reports, 2013. **3**: p. 2369.
23. Blasi, T., et al., *Label-free cell cycle analysis for high-throughput imaging flow cytometry*. Nat Commun, 2016. **7**: p. 10256.
24. Kobayashi, H., et al., *Label-free detection of cellular drug responses by high-throughput bright-field imaging and machine learning*. Sci Rep, 2017. **7**(1): p. 12454.
25. Wu, Y., et al., *Label-free bioaerosol sensing using mobile microscopy and deep learning*. ACS Photonics, 2018. **5**(11): p. 4617-4627.

26. Nitta, N., et al., *Intelligent image-activated cell sorting*. Cell, 2018. **175**(1): p. 266-276. e13.
27. Gorocs, Z., et al., *A deep learning-enabled portable imaging flow cytometer for cost-effective, high-throughput, and label-free analysis of natural water samples*. Light: Science & Applications, 2018.
28. Li, Y., et al., *Deep cytometry: deep learning with real-time inference in cell sorting and flow cytometry*. Scientific reports, 2019. **9**(1): p. 1-12.
29. Peuß, R., et al., *Label-independent flow cytometry and unsupervised neural network method for de novo clustering of cell populations*. bioRxiv, 2019: p. 603035.
30. Nketia, T.A., et al., *Analysis of live cell images: Methods, tools and opportunities*. Methods, 2017. **115**: p. 65-79.
31. DiGiorgio, C.L., D.A. Gonzalez, and C.C. Huitt, *Cryptosporidium and Giardia recoveries in natural waters by using environmental protection agency method 1623*. Applied and Environmental Microbiology, 2002. **68**(12): p. 5952-5955.
32. Vesey, G., et al., *Application of flow cytometric methods for the routine detection of Cryptosporidium and Giardia in water*. Cytometry: The Journal of the International Society for Analytical Cytology, 1994. **16**(1): p. 1-6.
33. Han, Y., et al., *Imaging technologies for flow cytometry*. Lab on a Chip, 2016. **16**(24): p. 4639-4647.

34. Ruske, S., et al., *Machine learning for improved data analysis of biological aerosol using the WIBS*. Atmospheric Measurement Techniques, 2018. **11**(11): p. 6203-6230.
35. Rösch, P., et al., *On-line monitoring and identification of bioaerosols*. Analytical Chemistry, 2006. **78**(7): p. 2163-2170.
36. Meng, N., et al., *Large-scale multi-class image-based cell classification with deep learning*. IEEE Journal of Biomedical and Health Informatics, 2018. **23**(5): p. 2091-2098.
37. Göröcs, Z., et al., *A deep learning-enabled portable imaging flow cytometer for cost-effective, high-throughput, and label-free analysis of natural water samples*. Light: Science & Applications, 2018. **7**(1): p. 1-12.
38. Zhang, Y., et al., *Motility-based label-free detection of parasites in bodily fluids using holographic speckle analysis and deep learning*. Light: Science & Applications, 2018. **7**(1): p. 1-18.
39. Kim, G., et al., *Learning-based screening of hematologic disorders using quantitative phase imaging of individual red blood cells*. Biosensors and Bioelectronics, 2019. **123**: p. 69-76.
40. Huang, G., et al. *Densely connected convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
41. Masana, M., et al., *Metric learning for novelty and anomaly detection*. arXiv preprint arXiv:1808.05492, 2018.
42. Barteneva, N.S. and I.A. Vorobjev, *Imaging Flow Cytometry*. 2016: Springer.

43. Erdbrugger, U., S. La Salvia, and J. Lannigan, *Detection of Extracellular Vesicles Using the ImageStream® X MKII Imaging Flow Cytometer*.
44. Franklin, D., *Nvidia Jetson TX2 delivers twice the intelligence to the edge*. NVIDIA Accelerated Computing—Parallel Forall, 2017.
45. Basiji, D.A., et al., *Cellular image analysis and imaging by flow cytometry*. Clin Lab Med, 2007. **27**(3): p. 653-70, viii.
46. Bonner, W.A., et al., *Fluorescence activated cell sorting*. Review of Scientific Instruments, 1972. **43**(3): p. 404-409.
47. Shi, Y.Z., et al., *Sculpting nanoparticle dynamics for single-bacteria-level screening and direct binding-efficiency measurement*. Nature communications, 2018. **9**(1): p. 1-11.
48. Lumini, A., L. Nanni, and G. Maguolo, *Deep learning for plankton and coral classification*. Applied Computing and Informatics, 2019.
49. Pedreira, C.E., et al., *Overview of clinical flow cytometry data analysis: recent advances and future challenges*. Trends in biotechnology, 2013. **31**(7): p. 415-425.
50. Wang, Y., et al., *Past, present and future applications of flow cytometry in aquatic microbiology*. Trends in biotechnology, 2010. **28**(8): p. 416-424.
51. Delaat, A., W. Gohde, and M. Vogelzack, *Determination of ploidy of single plants and plant populations by flow cytometry*. Plant breeding, 1987. **99**(4): p. 303-307.
52. Batchelor, B.G., *Machine Vision Handbook*. 2012: Springer.

53. Nixon, M. and A. Aguado, *Feature extraction and image processing for computer vision*. 2019: Academic press.
54. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. 2005. IEEE Computer Society.
55. Satpathy, A., X. Jiang, and H.L. Eng, *Human detection by quadratic classification on subspace of extended histogram of gradients*. IEEE Trans Image Process, 2014. **23**(1): p. 287-97.
56. Lowe, D.G., *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision, 2004. **60**(2): p. 91-110.
57. Ren, J., X. Jiang, and J. Yuan, *Noise-resistant local binary pattern with an embedded error-correction mechanism*. IEEE Trans Image Process, 2013. **22**(10): p. 4049-60.
58. Satpathy, A., X. Jiang, and H.L. Eng, *LBP-based edge-texture features for object recognition*. IEEE Trans Image Process, 2014. **23**(5): p. 1953-64.
59. Natekin, A. and A. Knoll, *Gradient boosting machines, a tutorial*. Frontiers in neurorobotics, 2013. **7**: p. 21.
60. Biau, G. and E. Scornet, *A random forest guided tour*. Test, 2016. **25**(2): p. 197-227.
61. Hennig, H., et al., *An open-source solution for advanced imaging flow cytometry data analysis using machine learning*. Methods, 2017. **112**: p. 201-210.

62. Nassar, M., et al., *Label-free identification of white blood cells using machine learning*. Cytometry Part A, 2019. **95**(8): p. 836-842.
63. Scholkopf, B., et al., *Estimating the support of a high-dimensional distribution*. Neural Comput, 2001. **13**(7): p. 1443-71.
64. Jiang, X., *Linear subspace learning-based dimensionality reduction*. IEEE Signal Processing Magazine, 2011. **28**(2): p. 16-26.
65. De Maesschalck, R., D. Jouan-Rimbaud, and D.L. Massart, *The mahalanobis distance*. Chemometrics and Intelligent Laboratory Systems, 2000. **50**(1): p. 1-18.
66. Peterson, M.A. and B.S. Gibson, *Shape recognition inputs to figure-ground organization in three-dimensional displays*. Cognitive Psychology, 1993. **25**(3): p. 383-429.
67. Turk, M.A. and A.P. Pentland. *Face recognition using eigenfaces*. in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*. 1991. IEEE Computer Society.
68. Greanias, E.C., et al., *The recognition of handwritten numerals by contour analysis*. IBM Journal of Research and Development, 1963. **7**(1): p. 14-21.
69. Ferryman, J.M., et al. *A Generic Deformable Model for Vehicle Recognition*. in *BMVC*. 1995. Citeseer.
70. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. 2012.

71. Zhao, Z.-Q., et al., *Object detection with deep learning: A review*. IEEE transactions on neural networks and learning systems, 2019. **30**(11): p. 3212-3232.
72. Noda, K., et al., *Audio-visual speech recognition using deep learning*. Applied Intelligence, 2015. **42**(4): p. 722-737.
73. Wainberg, M., et al., *Deep learning in biomedicine*. Nature biotechnology, 2018. **36**(9): p. 829-838.
74. Young, T., et al., *Recent trends in deep learning based natural language processing*. IEEE Computational intelligence magazine, 2018. **13**(3): p. 55-75.
75. Goodfellow, I., Y. Bengio, and A. Courville, *Deep learning*. 2016: MIT press.
76. Fukushima, K. and S. Miyake, *Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition*, in *Competition and cooperation in neural nets*. 1982, Springer. p. 267-285.
77. Ruder, S., *An overview of gradient descent optimization algorithms*. arXiv preprint arXiv:1609.04747, 2016.
78. Ciresan, D.C., et al., *Deep, big, simple neural nets for handwritten digit recognition*. Neural computation, 2010. **22**(12): p. 3207-3220.
79. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
80. Szegedy, C., et al. *Going deeper with convolutions*. 2015.

81. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
82. Sandler, M., et al., *Mobilenetv2: Inverted residuals and linear bottlenecks*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. **1**(1): p. 4510-4520.
83. Hu, J., L. Shen, and G. Sun. *Squeeze-and-excitation networks*. 2018.
84. Kolesnikov, A., et al., *Big Transfer (BiT): General Visual Representation Learning Supplementary Material*.
85. Srivastava, R.K., K. Greff, and J. Schmidhuber, *Highway networks*. arXiv preprint arXiv:1505.00387, 2015.
86. Howard, A.G., et al., *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.04861, 2017.
87. Huang, G., et al. *Condensenet: An efficient densenet using learned group convolutions*. 2018.
88. Chollet, F., *Xception: Deep Learning with Depthwise Separable Convolutions*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: p. 1800-1807.
89. Luo, S., et al., *Rare bioparticle detection via deep metric learning*. *RSC Advances*, 2021. **11**(29): p. 17603-17610.
90. Lu, J., J. Hu, and J. Zhou, *Deep metric learning for visual understanding: An overview of recent advances*. *IEEE Signal Processing Magazine*, 2017. **34**(6): p. 76-84.

91. Hoffer, E. and N. Ailon. *Deep metric learning using triplet network*. in *International workshop on similarity-based pattern recognition*. 2015. Springer.
92. Hain, R., C.J. Kähler, and C. Tropea, *Comparison of CCD, CMOS and intensified cameras*. *Experiments in fluids*, 2007. **42**(3): p. 403-411.
93. McClish, M., et al., *Recent advances of planar silicon APD technology*. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 2006. **567**(1): p. 36-40.
94. Basiji, D.A., et al., *Cellular image analysis and imaging by flow cytometry*. *Clinics in laboratory medicine*, 2007. **27**(3): p. 653-670.
95. Elliott, G.S., *Moving pictures: imaging flow cytometry for drug development*. *Combinatorial chemistry & high throughput screening*, 2009. **12**(9): p. 849-859.
96. Basiji, D.A., *Principles of Amnis imaging flow cytometry*, in *Imaging Flow Cytometry*. 2016, Springer. p. 13-21.
97. Schonbrun, E., S.S. Gorthi, and D. Schaak, *Microfabricated multiple field of view imaging flow cytometry*. *Lab on a Chip*, 2012. **12**(2): p. 268-273.
98. Gorthi, S.S., D. Schaak, and E. Schonbrun, *Fluorescence imaging of flowing cells using a temporally coded excitation*. *Optics express*, 2013. **21**(4): p. 5164-5170.
99. Diebold, E.D., et al., *Digitally synthesized beat frequency multiplexing for sub-millisecond fluorescence microscopy*. *Nature Photonics*, 2013. **7**(10): p. 806-810.

100. Mikami, H., et al. *Ultrafast Confocal Fluorescence Microscopy by Frequency-Division-Multiplexed Multi-Line Focusing*. in *CLEO: Applications and Technology*. 2016. Optical Society of America.
101. Han, Y. and Y.-H. Lo, *Imaging cells in flow cytometer using spatial-temporal transformation*. *Scientific reports*, 2015. **5**: p. 13267.
102. Han, Y. and Y.-H. Lo. *Imaging flow cytometer using computation and spatially coded filter*. in *High-Speed Biomedical Imaging and Spectroscopy: Toward Big Data Instrumentation and Management*. 2016. International Society for Optics and Photonics.
103. Sieracki, C.K., M.E. Sieracki, and C.S. Yentsch, *An imaging-in-flow system for automated analysis of marine microplankton*. *Marine Ecology Progress Series*, 1998. **168**: p. 285-296.
104. Poulton, N.J., *FlowCam: quantification and classification of phytoplankton by imaging flow cytometry*, in *Imaging Flow Cytometry*. 2016, Springer. p. 237-247.
105. Olson, R.J. and H.M. Sosik, *A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging FlowCytobot*. *Limnology and Oceanography: Methods*, 2007. **5**(6): p. 195-203.
106. Luo, S., et al., *Deep learning-enabled imaging flow cytometry for high-speed Cryptosporidium and Giardia detection*. *Cytometry Part A*. **n/a**(n/a).
107. Luo, S., et al., *Machine Learning-Based Pipeline for High Accuracy Bioparticle Sizing*. *Micromachines*, 2020. **11**(12): p. 1084.

108. Ciresan, D.C., et al., *Mitosis detection in breast cancer histology images with deep neural networks*. *Med Image Comput Comput Assist Interv*, 2013. **16**(Pt 2): p. 411-8.
109. Hung, J. and A. Carpenter. *Applying faster R-CNN for object detection on malaria images*. 2017.
110. Chen, L.C., et al., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. *IEEE Trans Pattern Anal Mach Intell*, 2018. **40**(4): p. 834-848.
111. He, K., et al., *Mask R-CNN*. *IEEE Trans Pattern Anal Mach Intell*, 2018.
112. Ning, G., et al. *Spatially supervised recurrent convolutional neural networks for visual object tracking*. in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017. IEEE.
113. Wang, Q., et al. *Fast online object tracking and segmentation: A unifying approach*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019.
114. Wojke, N., A. Bewley, and D. Paulus. *Simple online and realtime tracking with a deep association metric*. in *2017 IEEE international conference on image processing (ICIP)*. 2017. IEEE.
115. Voigtlaender, P., et al. *MOTS: Multi-object tracking and segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019.

116. Li, Y., L. Shen, and S. Yu, *HEp-2 Specimen Image Segmentation and Classification Using Very Deep Fully Convolutional Network*. IEEE Trans Med Imaging, 2017. **36**(7): p. 1561-1572.
117. Shelhamer, E., J. Long, and T. Darrell, *Fully Convolutional Networks for Semantic Segmentation*. IEEE Trans Pattern Anal Mach Intell, 2017. **39**(4): p. 640-651.
118. Mikami, H., et al., *Virtual-freezing fluorescence imaging flow cytometry*. Nature communications, 2020. **11**(1): p. 1-11.
119. Doan, M., et al., *Label-free leukemia monitoring by computer vision*. Cytometry Part A, 2020. **97**(4): p. 407-414.
120. Correa, I., et al. *Deep learning for microalgae classification*. in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017. IEEE.
121. Lumini, A. and L. Nanni, *Deep learning and transfer learning features for plankton classification*. Ecological informatics, 2019. **51**: p. 33-43.
122. Lumini, A., L. Nanni, and G. Maguolo, *Deep learning for plankton and coral classification*. Applied Computing and Informatics, 2020.
123. Liu, J., et al. *Deep pyramidal residual networks for plankton image classification*. in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*. 2018. IEEE.
124. Qin, X. and Z. Wang, *NASNet: A Neuron Attention Stage-by-Stage Net for Single Image Deraining*. arXiv preprint arXiv:1912.03151, 2019.

125. Han, D., J. Kim, and J. Kim. *Deep pyramidal residual networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
126. Eulenberg, P., et al., *Reconstructing cell cycle and disease progression using deep learning*. *Nature communications*, 2017. **8**(1): p. 1-6.
127. Lei, C., et al., *High-throughput imaging flow cytometry by optofluidic time-stretch microscopy*. *Nature protocols*, 2018. **13**(7): p. 1603-1631.
128. Gupta, A., et al., *Deep learning in image cytometry: a review*. *Cytometry Part A*, 2019. **95**(4): p. 366-380.
129. Sun, J., A. Tárnok, and X. Su, *Deep learning-based single-cell optical image studies*. *Cytometry Part A*, 2020. **97**(3): p. 226-240.
130. Goda, K., et al., *High-throughput single-microparticle imaging flow analyzer*. *Proceedings of the National Academy of Sciences*, 2012. **109**(29): p. 11630-11635.
131. Lippeveld, M., et al., *Classification of human white blood cells using machine learning for stain-free imaging flow cytometry*. *Cytometry Part A*, 2020. **97**(3): p. 308-319.
132. Millipore, A.p.o.E. *IDEAS Image Data Exploration and Analysis Software User's Manual*. 2015.
133. Dreelin, E.A., et al., *Cryptosporidium and Giardia in surface water: a case study from Michigan, USA to inform management of rural water systems*. *International journal of environmental research and public health*, 2014. **11**(10): p. 10480-10503.

134. Shifat-E-Rabbi, M., et al., *Cell image classification: a comparative overview*. Cytometry Part A, 2020. **97**(4): p. 347-362.
135. Erdbrügger, U., et al., *Imaging flow cytometry elucidates limitations of microparticle analysis by conventional flow cytometry*. Cytometry Part A, 2014. **85**(9): p. 756-770.
136. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning*. Nature, 2015. **521**(7553): p. 436-44.
137. Corporation, L. *Amnis® AI Image Analysis Software*. [cited 2021 25 Jan]; Available from: <https://www.luminexcorp.com/download/amnis-ai-image-analysis-software/>.
138. Park, J., et al., *Algal morphological identification in watersheds for drinking water supply using neural architecture search for convolutional neural network*. Water, 2019. **11**(7): p. 1338.
139. Dunker, S., et al., *Combining high-throughput imaging flow cytometry and deep learning for efficient species and life-cycle stage identification of phytoplankton*. BMC ecology, 2018. **18**(1): p. 51.
140. Liu, P., et al., *Analysis and prediction of water quality using LSTM deep neural networks in IoT environment*. Sustainability, 2019. **11**(7): p. 2058.
141. Bürmen, M., F. Pernuš, and B. Likar, *LED light sources: a survey of quality-affecting factors and methods for their assessment*. Measurement science and technology, 2008. **19**(12): p. 122002.

142. Fossum, E.R. and D.B. Hondongwa, *A review of the pinned photodiode for CCD and CMOS image sensors*. IEEE Journal of the Electron Devices Society, 2014.
143. Rose, J.B. and T.R. Slifko, *Giardia, Cryptosporidium, and Cyclospora and their impact on foods: a review*. Journal of food protection, 1999. **62**(9): p. 1059-1070.
144. Sato, M.I.Z., et al., *Assessing the infection risk of Giardia and Cryptosporidium in public drinking water delivered by surface water systems in Sao Paulo State, Brazil*. Science of The Total Environment, 2013. **442**: p. 389-396.
145. Koelmans, A.A., et al., *Microplastics in freshwaters and drinking water: critical review and assessment of data quality*. Water research, 2019. **155**: p. 410-422.
146. LeCun, Y., et al. *Handwritten digit recognition with a back-propagation network*. in *Advances in Neural Information Processing Systems*. 1990.
147. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python*. the Journal of machine Learning research, 2011. **12**: p. 2825-2830.
148. Abadi, M., et al. *Tensorflow: A system for large-scale machine learning*. in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016.
149. Brown, L., *Deep learning with GPUs*. Larry Brown Ph. D., Johns Hopkins University, 2015.

150. Glorot, X. and Y. Bengio. *Understanding the difficulty of training deep feedforward neural networks*. in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010.
151. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
152. Perez, L. and J. Wang, *The effectiveness of data augmentation in image classification using deep learning*. arXiv preprint arXiv:1712.04621, 2017.
153. Bekkar, M. and T.A. Alitouche, *Imbalanced data learning approaches review*. International Journal of Data Mining & Knowledge Management Process, 2013. **3**(4): p. 15.
154. Tan, M. and Q.V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv preprint arXiv:1905.11946, 2019.
155. Van Asch, V., *Macro-and micro-averaged evaluation measures [[basic draft]]*. Belgium: CLiPS, 2013. **49**.
156. Maaten, L.v.d. and G. Hinton, *Visualizing data using t-SNE*. Journal of Machine Learning Research, 2008. **9**(Nov): p. 2579-2605.
157. Dunteman, G.H., *Principal components analysis*. 1989: Sage.
158. Caruana, R., *Multitask learning*. Machine Learning, 1997. **28**(1): p. 41-75.
159. Chen, Y., et al., *Rare cell isolation and analysis in microfluidics*. Lab on a Chip, 2014. **14**(4): p. 626-645.
160. Goodfellow, I., et al., *Deep learning*. Vol. 1. 2016: MIT press Cambridge.

161. Reichardt, T.A., et al. *Analysis of flow-cytometer scattering and fluorescence data to identify particle mixtures*. in *Optics and Photonics in Global Homeland Security IV*. 2008. International Society for Optics and Photonics.
162. Bendale, A. and T. Boulton. *Towards open world recognition*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
163. Pimentel, M.A., et al., *A review of novelty detection*. Signal Processing, 2014. **99**: p. 215-249.
164. Bendale, A. and T.E. Boulton. *Towards open set deep networks*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
165. Hendrycks, D. and K. Gimpel, *A baseline for detecting misclassified and out-of-distribution examples in neural networks*. arXiv preprint arXiv:1610.02136, 2016.
166. Meyer, B.J., B. Harwood, and T. Drummond. *Deep metric learning and image classification with nearest neighbour gaussian kernels*. in *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018. IEEE.
167. Trigueros, D.S., L. Meng, and M. Hartnett, *Face recognition: From traditional to deep learning methods*. arXiv preprint arXiv:1811.00116, 2018.
168. Bromley, J., et al. *Signature verification using a " siamese" time delay neural network*. in *Advances in Neural Information Processing Systems*. 1994.
169. Taigman, Y., et al. *Deepface: Closing the gap to human-level performance in face verification*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

170. Varior, R.R., M. Haloi, and G. Wang. *Gated siamese convolutional neural network architecture for human re-identification*. in *European Conference on Computer Vision*. 2016. Springer.
171. He, K., et al. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
172. Paszke, A., et al. *Pytorch: An imperative style, high-performance deep learning library*. in *Advances in Neural Information Processing Systems*. 2019.
173. Caruana, R., S. Lawrence, and C.L. Giles. *Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping*. in *Advances in Neural Information Processing Systems*. 2001.
174. Reynolds, D.A., *Gaussian Mixture Models*. Encyclopedia of Biometrics, 2009. **741**.
175. Mendenhall, W.M. and T.L. Sincich, *Statistics for Engineering and the Sciences*. 2016: CRC Press.
176. Corduneanu, A. and C.M. Bishop. *Variational Bayesian model selection for mixture distributions*. in *Artificial Intelligence and Statistics*. 2001. Morgan Kaufmann Waltham, MA.
177. Forsyth, D.A. and J. Ponce, *Computer vision: a modern approach*. 2012: Pearson.
178. Mage, P.L., et al., *Shape-based separation of synthetic microparticles*. Nature materials, 2019. **18**(1): p. 82

179. Park, J.H., et al., *Inflated sporopollenin exine capsules obtained from thin-walled pollen*. Scientific reports, 2016. **6**(1): p. 1-10.
180. Viles, C.L. and M.E. Sieracki, *Measurement of marine picoplankton cell size by using a cooled, charge-coupled device camera with image-analyzed fluorescence microscopy*. Applied and environmental microbiology, 1992. **58**(2): p. 584-592.
181. Medema, G.J., et al., *Sedimentation of Free and Attached Cryptosporidium Oocysts and Giardia Cysts in Water*. Appl. Environ. Microbiol., 1998. **64**(11): p. 4460-4466.
182. Abbireddy, C.O. and C.R. Clayton, *A review of modern particle sizing methods*. Proceedings of the Institution of Civil Engineers-Geotechnical Engineering, 2009. **162**(4): p. 193-201.
183. Costa, C., et al., *Extracting fish size using dual underwater cameras*. Aquacultural Engineering, 2006. **35**(3): p. 218-227.
184. Almeida, S.M., et al., *Approaching PM_{2.5} and PM_{2.5-10} source apportionment by mass balance analysis, principal component analysis and particle size distribution*. Science of the Total Environment, 2006. **368**(2-3): p. 663-674.
185. Fernlund, J.M., *The effect of particle form on sieve analysis: a test by image analysis*. Engineering Geology, 1998. **50**(1-2): p. 111-124.

186. Saveyn, H., et al., *In-line comparison of particle sizing by static light scattering, time-of-transition, and dynamic image analysis*. Particle & Particle Systems Characterization, 2006. **23**(2): p. 145-153.
187. Brown, W., *Dynamic light scattering: the method and some applications*. Vol. 313. 1993: Clarendon Press Oxford.
188. Filipe, V., A. Hawe, and W. Jiskoot, *Critical evaluation of Nanoparticle Tracking Analysis (NTA) by NanoSight for the measurement of nanoparticles and protein aggregates*. Pharmaceutical research, 2010. **27**(5): p. 796-810.
189. Weiner, B., W.W. Tscharnuter, and N. Karasikov, *Improvements in Accuracy and Speed Using the Time-of-Transition Method and Dynamic Image Analysis for Particle Sizing: Some Real-World Examples*. 1998, ACS Publications.
190. Bradbury, S., et al., *Introduction to light microscopy*. 1998: Bios Scientific Oxford, UK.
191. Hamilton, N., *Quantification and its applications in fluorescent microscopy imaging*. Traffic, 2009. **10**(8): p. 951-961.
192. Shi, Y., et al., *Chirality-assisted lateral momentum transfer for bidirectional enantioselective separation*. Light: Science & Applications, 2020. **9**(1): p. 1-12.
193. Flegler, S.L. and S.L. Flegler, *Scanning & Transmission Electron Microscopy*. 1997: Oxford University Press.
194. Shi, Y., et al., *Nanophotonic array-induced dynamic behavior for label-free shape-selective bacteria sieving*. ACS nano, 2019. **13**(10): p. 12070-12080.

195. Lawrence, W.G., et al. *A comparison of avalanche photodiode and photomultiplier tube detectors for flow cytometry*. in *Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues VI*. 2008. International Society for Optics and Photonics.
196. Han, Y., et al., *Review: imaging technologies for flow cytometry*. *Lab Chip*, 2016. **16**(24): p. 4639-4647.
197. Pham, H., *Springer handbook of engineering statistics*. 2006: Springer Science & Business Media.
198. Van Valen, D.A., et al., *Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments*. *PLoS computational biology*, 2016. **12**(11): p. e1005177.
199. Girshick, R. *Fast r-cnn*. in *Proceedings of the IEEE international conference on computer vision*. 2015.
200. Lin, T.-Y., et al. *Focal loss for dense object detection*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
201. Moen, E., et al., *Deep learning for cellular image analysis*. *Nature methods*, 2019: p. 1-14.
202. Szeliski, R., *Computer Vision: Algorithms and Applications*. Instructor, 2019. **201901**.
203. Canny, J., *A computational approach to edge detection*. *IEEE Transactions on pattern analysis and machine intelligence*, 1986(6): p. 679-698.

204. Gil, J.Y. and R. Kimmel, *Efficient dilation, erosion, opening, and closing algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **24**(12): p. 1606-1617.
205. Suzuki, S., *Topological structural analysis of digitized binary images by border following*. Computer vision, graphics, and image processing, 1985. **30**(1): p. 32-46.
206. Udupa, J.K., et al. *Methodology for evaluating image-segmentation algorithms*. 2002. International Society for Optics and Photonics.
207. Chai, T. and R.R. Draxler, *Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature*. Geoscientific model development, 2014. **7**(3): p. 1247-1250.
208. Cheng, J., et al., *Recent advances in efficient computation of deep convolutional neural networks*. Frontiers of Information Technology & Electronic Engineering, 2018. **19**(1): p. 64-77.
209. Yang, Z., et al., *Searching for low-bit weights in quantized neural networks*. arXiv preprint arXiv:2009.08695, 2020.
210. Han, S., et al., *Learning both weights and connections for efficient neural networks*. arXiv preprint arXiv:1506.02626, 2015.
211. Han, S., H. Mao, and W.J. Dally, *Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding*. arXiv preprint arXiv:1510.00149, 2015.

212. Kolda, T.G. and B.W. Bader, *Tensor decompositions and applications*. SIAM review, 2009. **51**(3): p. 455-500.
213. Zhang, X., et al., *Accelerating very deep convolutional networks for classification and detection*. IEEE transactions on pattern analysis and machine intelligence, 2015. **38**(10): p. 1943-1955.
214. Kim, Y.-D., et al., *Compression of deep convolutional neural networks for fast and low power mobile applications*. arXiv preprint arXiv:1511.06530, 2015.
215. Denil, M., et al., *Predicting parameters in deep learning*. arXiv preprint arXiv:1306.0543, 2013.
216. Chen, W., et al. *Compressing neural networks with the hashing trick*. in *International conference on machine learning*. 2015. PMLR.
217. Miyashita, D., E.H. Lee, and B. Murmann, *Convolutional neural networks using logarithmic data representation*. arXiv preprint arXiv:1603.01025, 2016.
218. Hammerstrom, D. *A VLSI architecture for high-performance, low-cost, on-chip learning*. in *1990 IJCNN International Joint Conference on Neural Networks*. 1990. IEEE.
219. Chen, G., et al. *Learning efficient object detection models with knowledge distillation*. in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017.
220. Hinton, G., O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*. arXiv preprint arXiv:1503.02531, 2015.

221. Patel, A.B., T. Nguyen, and R.G. Baraniuk, *A probabilistic theory of deep learning*. arXiv preprint arXiv:1504.00641, 2015.
222. Tran, D., et al., *Deep probabilistic programming*. arXiv preprint arXiv:1701.03757, 2017.
223. Depeweg, S., *Modeling Epistemic and Aleatoric Uncertainty with Bayesian Neural Networks and Latent Variables*. 2019, Technische Universität München.

Author's Publications

Journal Publications

1. **S. Luo**, K. T. Nguyen, B. T. T. Nguyen, S. Feng, Y. Shi, A. Elsayed, Y. Zhang, X. Zhou, B. Wen, G. Chierchia, H. Talbot, T. Bourouina, X. Jiang, and A. Q. Liu. "Deep Learning-Enabled Imaging Flow Cytometry for High-Speed *Cryptosporidium* and *Giardia* Detection". *Cytometry Part A*, 2021. 10.1002/cyto.a.24321.
2. **S. Luo**, Y. Zhang, K. T. Nguyen, S.L. Feng, H. Yu, G. Chierchia, H. Talbot, X. Jiang, T. Bourouina, A. Q. Liu. "Machine Learning-Based Pipeline for High Accuracy Bioparticle Sizing". *Micromachine*, 2020. **11**(12): p.1084.
3. **S. Luo**, Y. Shi, L. K. Chin, Y. Zhang, B. Wen, Y. Sun, G. Chierchia, H. Talbot, X. Jiang, T. Bourouina, A. Q. Liu. "Rare Bioparticle Detection via Deep Metric Learning". *RSC Advances*, 2021; **11**:17603-17610
4. **S. Luo**, Y. Shi, L. K. Chin, P. Hutchinson, Y. Zhang, G. Chierchia, H. Talbot, X. Jiang, T. Bourouina, and A. Q. Liu. "Machine Learning Assisted Intelligent Imaging Flow Cytometry: A Review". *Advanced Intelligent System*, 2021 (Accepted).
5. Z. Li, H. Zhang, B.T.T. Nguyen, **S. Luo**, Y. Liu, J. Zou, Y. Shi, H. Cai, Z. Yang, Y. Jin, Y. Hao, Y. Zhang, and A. Q. Liu. "Smart Ring Resonator-based Sensor for Multicomponent Chemical Analysis via Machine Learning". *Photonics Research*, 2020. **9**(2):B38-44
6. A. Elsayed, M. Erfan, Y. M. Sabry, R. Dris, J. Gasperi, J. Barbier, F. Marty, F. Bouanis, **S. Luo**, B. T. T. Nguyen, A. Q. Liu, B. Tassin, T. Bourouina, "A microfluidic chip enables fast analysis of water microplastics by optical spectroscopy". *Scientific Reports*, 2021. **11**: 10533.

Conference Presentations

1. **S. Luo**, K. T. Nguyen, X. Jiang, J. Wu, B. Wen, Y. Zhang, G. Chierchia, H. Talbot, T. Bourouina, A. Q. Liu, (2019, July). “A High Performance of Single Cell Imaging Detection with Deep Learning”. In *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)* (pp. 356-360). IEEE.
(Oral)

Awards

1. System Design Contest Second Place (Embedded System Implementation of Neural Network based Object Detection for Drones), *IEEE/ACM Design Automation Conference*, 2020.

**Apprentissage automatique sur la reconnaissance et
le dimensionnement d'images de bioparticules à
haut débit**

La cytométrie de flux est l'une des technologies les plus largement adoptées pour l'analyse des bioparticules dans le diagnostic des maladies [9-11], l'inspection des aliments [12, 13] et la surveillance de la qualité de l'eau [14-16] parmi d'autres applications de détection de bioparticules [17-21]. La cytométrie en flux conventionnelle identifie les bioparticules en fonction des signatures optiques des bioparticules telles que leurs profils fluorescents [17]. Les bioparticules telles que les cellules sont généralement immuno-marquées avec des marqueurs fluorescents qui ciblent des biomarqueurs spécifiques aux cellules pour faciliter l'analyse. Le marquage fluorescent peut toutefois altérer les propriétés cellulaires et interférer avec les analyses en aval [22], et aucune connaissance de la morphologie cellulaire n'a pu être dérivée des données rares obtenues à l'aide de la cytométrie en flux conventionnelle.

La morphologie est une caractéristique importante des bioparticules. Les images regorgent d'informations riches sur la morphologie qui pourraient être utilisées pour la détection des bioparticules [15-22]. Les méthodes de référence reposent principalement sur la morphologie des bioparticules avec des techniques d'imagerie basées sur la microscopie pour l'identification microbienne [23]. Par exemple, les protozoaires pathogènes dans l'eau potable, *Cryptosporidium* et *Giardia*, sont détectés en analysant la morphologie des images microscopiques [24]. Cependant, l'approche d'imagerie traditionnelle basée sur la microscopie nécessite une préparation intensive des échantillons et une analyse manuelle des données, ce qui entrave considérablement son application dans l'analyse de bioparticules à haut débit [25].

La cytométrie en flux basée sur l'imagerie est une technologie d'acquisition d'images à haut débit en couplant des techniques d'imagerie à la cytométrie en flux.

La cytométrie en flux d'imagerie a un débit significativement plus élevé que les techniques d'imagerie traditionnelles basées sur la microscopie et est capable de capturer des milliers à des millions d'images de bioparticules en une seconde en utilisant diverses modalités d'imagerie [33]. Les images acquises regorgent d'informations riches sur la morphologie des cellules et des microparticules. Néanmoins, le grand nombre d'images riches en informations pose un grand défi pour l'analyse des données.

L'apprentissage automatique est souvent utilisé pour identifier les bioparticules telles que les cellules, les planctons et les micro-algues lors de l'analyse d'images [34, 35]. À l'exception de quelques cas où l'apprentissage automatique est utilisé pour la correction, l'amélioration et la reconstruction d'image [27], un certain nombre d'algorithmes d'apprentissage automatique ont été développés pour la détection et la classification des cellules basées sur l'imagerie [5, 25, 36-39] avec une précision élevée dans ces travaux, mais ils étaient relativement lents et nécessitaient des ressources de calcul élevées au-delà de la capacité des systèmes embarqués afin de construire une machine de cytométrie en flux d'imagerie abordable. Par exemple, un algorithme très précis basé sur un réseau neuronal sophistiqué et densément connecté [40] pour la détection de cellules cancéreuses à haut débit dans le sang a été rapporté [5]. Il avait de bonnes performances de détection, mais sa vitesse était limitée à 100 fps (*frames per second*), et il devait s'appuyer sur une plate-forme haute performance avec un GPU Nvidia GeForce GTX 1080Ti haut de gamme. Le tri intelligent de cellules activé par l'image 2.0 (*Intelligent image-activated cell sorting 2.0*) permet d'atteindre un débit plus élevé en triant jusqu'à 2000 événements par seconde, mais il nécessite un serveur de 8 PC avec 8 processeurs multi-cœurs et GPU (NVIDIA GeForce GTX 1080 Ti) pour le traitement d'image avec apprentissage profond [4].

La plupart des algorithmes d'apprentissage automatique développés pour l'imagerie de la cytométrie en flux se concentrent principalement sur la précision de détection, mais négligent l'efficacité en termes de vitesse et d'exigences matérielles, ce qui a des implications pratiques.

L'apprentissage profond avec un réseau neuronal profond a atteint des performances très élevées pour de nombreuses applications riches en données et en apprentissage supervisé [5, 25, 36-39]. Cependant, les réseaux neuronaux profonds de pointe rencontrent encore des défis dans des applications pratiques telles que le système d'alerte précoce [8]. Dans ces applications, le nombre de bioparticules cibles est extrêmement faible et le nombre d'images d'arrière-plan est extrêmement important. Par conséquent, les modèles doivent avoir une performance avec une faible fausse alarme ainsi qu'un taux de récupération élevé. De plus, les images des bioparticules cibles sont difficiles à collecter pour créer un ensemble de données de référence. Par exemple, l'apparence des images collectées des bioparticules dans les environnements de test peut être différente de celles de l'ensemble de données d'apprentissage. Ces différences introduisent des difficultés pour les reconnaître avec les réseaux de neurones profonds traditionnels étant donné que le recours aux réseaux de neurones profonds suppose un monde statique et fermé et ne peuvent pas bien fonctionner sur un problème à jeu ouvert, de sorte que l'entrée au moment de l'inférence doit appartenir à la même distribution pendant l'entraînement afin d'obtenir de bonnes performances de détection [41].

Le système d'imagerie offre une mesure de taille basée sur l'image ainsi qu'une analyse de bioparticules à haut débit [42]. Des micrographies riches en informations de particules individuelles sont acquises à une vitesse élevée, et la taille des particules est déterminée en convertissant le pixel en longueur à un taux de conversion fixe qui

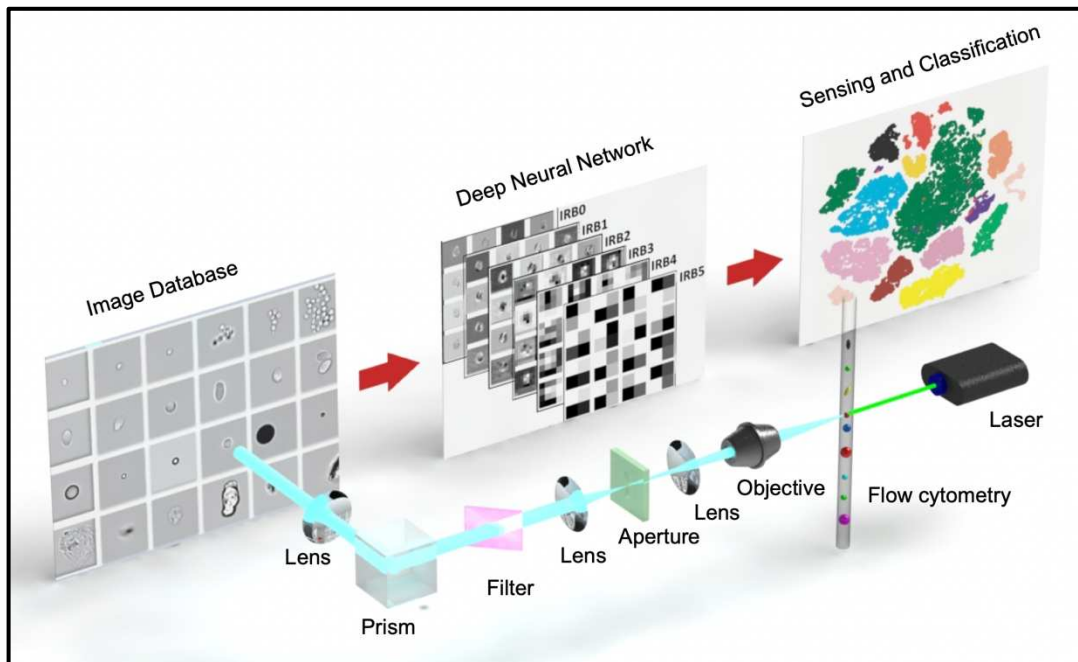


Figure 1: Vue d'ensemble de MCellNet, un réseau de neurones profonds qui permet l'imagerie par cytométrie en flux (Amnis® ImageStream®X Mk II) pour la détection de *Cryptosporidium* et *Giardia*. Le système comprend un laser, une cytométrie en flux, un système d'imagerie, une base de données d'images et un réseau neuronal profond.

est calculé théoriquement sur la base de la spécification des composants optiques. Par exemple, un seul pixel correspond à $0,33 \mu\text{m}$ dans le cytomètre de flux d'imagerie Amnis® ImageStream®X Mk II [43] avec un objectif de $60\times$ et $0,5 \mu\text{m}$ avec un objectif de $40\times$ selon les spécifications du fabricant. Cependant, nous avons remarqué que ce taux de conversion fixe ne conduit pas toujours à un dimensionnement précis des particules.

Face à ces défis, l'objectif majeur de cette thèse de doctorat est de développer les algorithmes d'apprentissage automatique pour détecter et dimensionner la bioparticule microscopique en utilisant l'apprentissage automatique, en particulier les méthodes d'apprentissage profond. Plus précisément, trois types de domaines sont explorés et discutés, à savoir : (1) détection de bioparticules à grande vitesse activée par l'apprentissage profond ; (2) Apprentissage métrique profond pour la détection de

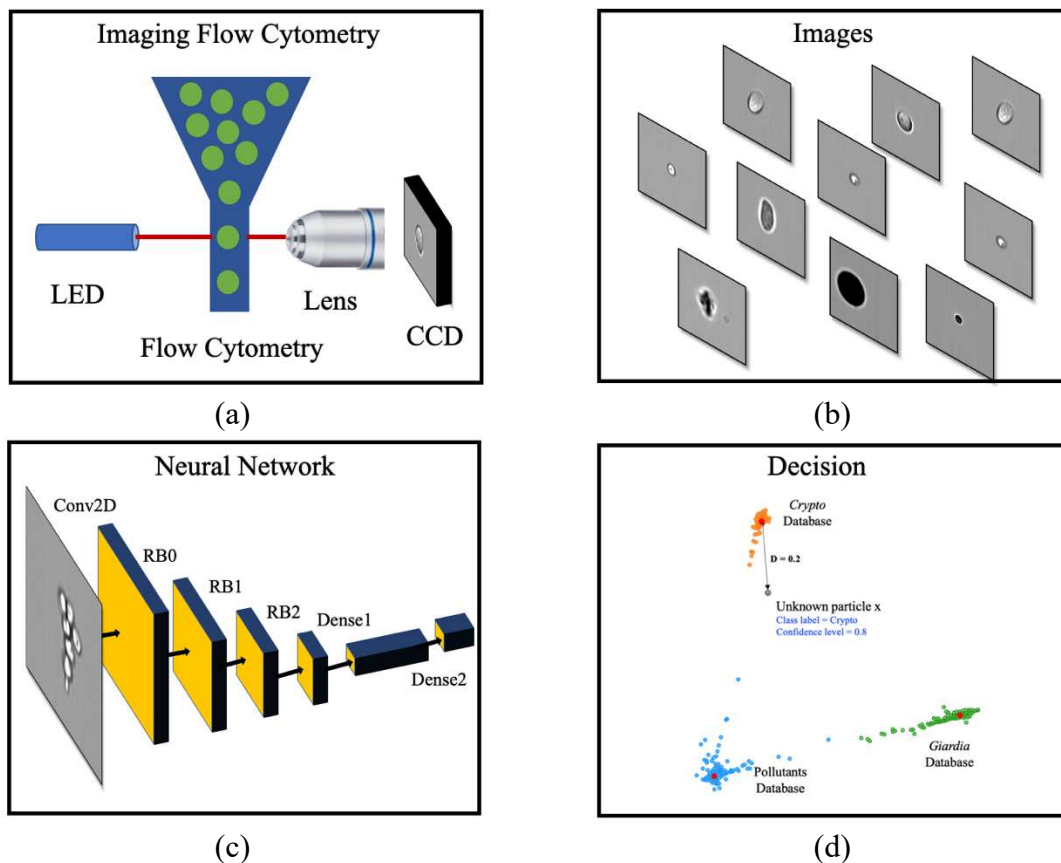


Figure 2: Vue d'ensemble de la configuration du système, un réseau neuronal métrique profond qui permet une cytométrie en flux d'imagerie intelligente (Amnis® ImageStream®X Mk II) pour la détection de bioparticules rares. Le système comprend (a) un système d'imagerie de cytométrie en flux, (b) une base de données d'images, (c) un réseau neuronal métrique profond et (d) un algorithme de décision, etc.

bioparticules rares ; (3) Pipeline basé sur l'apprentissage automatique pour la classification en taille des bioparticules.

Pour permettre l'inférence à faible puissance et à haut débit sur les périphériques embarqués à ressources limitées, par exemple Nvidia Jetson TX2 [44] à faible coût, un programme de détection en temps réel activé par l'apprentissage profond pour la détection de bioparticules à haut débit et sans marqueur fluorescent, comme le montre la **Figure 1**, a été proposé. Cette plate-forme de détection intelligente combine l'imagerie-cytométrie en flux et un réseau neuronal efficace que nous avons

développé, connu sous le nom de MCellNet, avec une cible optimisée pour la vitesse de traitement et la précision de détection. Nous rapportons un système à haut débit activé par l'apprentissage profond pour prédire *Cryptosporidium* et *Giardia* à l'aide de la cytométrie en flux par imagerie. Ce système combine l'algorithme d'apprentissage profond de pointe MobileNetV2 en vitesse (251 images par seconde) avec une précision de classification comparable. Le système signalé permet une détection de bioparticules rapide, précise et à haut débit dans les diagnostics cliniques, la surveillance environnementale et d'autres applications potentielles de biocapteur.

Pour traiter le problème de détection de bioparticules rares, un modèle de réseau neuronal profond basé sur l'apprentissage métrique comme le montre la **Figure 2** a été proposé. L'algorithme exploite le réseau de neurones convolutifs pour étudier les fonctionnalités riches de l'ensemble de données d'entraînement et apprend une métrique distincte en utilisant un réseau siamois et une fonction de perte contrastive qui a appris à maximiser la distance des différentes classes et à minimiser la distance des classes similaires. L'apprentissage métrique profond offre une possibilité de génération et utilise les informations de distance pour modéliser la similitude des images en apprenant une fonction mappe des pixels d'images à l'espace latent, jouant un rôle essentiel dans la détection d'objets rares. Dans cette partie, nous proposons un modèle robuste basé sur un réseau neuronal métrique profond pour la détection de bioparticules rares. Les expériences montrent que le réseau neuronal métrique profond a atteint une précision élevée de 99,86% dans la classification, 98,89% en taux de précision, 99,16% en taux de rappel et zéro taux de fausse alarme pour la détection de bioparticules rares. Le modèle rapporté permet à l'imagerie de cytométrie en flux dans le diagnostic biomédical, la surveillance environnementale et d'autres applications de biodétection.

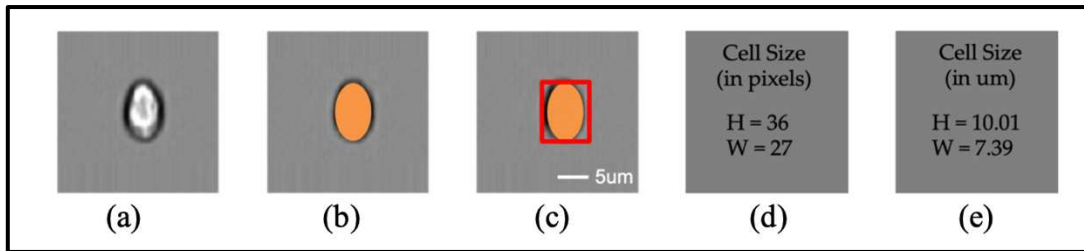


Figure 3 : Différentes étapes du pipeline de mesure de taille. (a) Image de bioparticules. (b) Segmentation. (c) Forme des bioparticules. (d) Taille des bioparticules en nombre de pixels. (e) Taille des bioparticules en μm . (a), (b) et (c) partagent la même barre d'échelle.

Pour permettre une mesure de la taille des bioparticules de haute précision, un pipeline basé sur l'apprentissage automatique pour le dimensionnement des bioparticules de haute précision basé sur l'imagerie, comme le montre la **Figure 3**, est démontré. Le pipeline se compose d'un module de segmentation d'image pour extraire les contours et estimer la taille de pixel de la bioparticule ainsi que d'un modèle d'apprentissage automatique pour une conversion pixel-taille précise. L'algorithme de segmentation d'image atteint 84,4% de l'IoU moyenne, et la taille de particule déterminée par le modèle d'apprentissage automatique n'a qu'un pourcentage d'erreur moyen de 4,2%, ce qui est cinq fois mieux que la méthode utilisant un rapport de conversion pixel-taille fixe (23,3%). Notre méthode permet à différents systèmes d'imagerie intelligents tels que l'imagerie de cytométrie en flux pour le dimensionnement des particules de haute précision et promet un grand potentiel pour une large gamme d'applications dans le domaine de la détection environnementale, du diagnostic biomédical et de la caractérisation des matériaux.