

Large-Scale Multi-Class Image-Based Cell Classification With Deep Learning

Nan Meng¹, Edmund Y. Lam¹, *Fellow, IEEE*, Kevin K. Tsia, *Member, IEEE*, and Hayden Kwok-Hay So¹, *Senior Member, IEEE*

Abstract—Recent advances in ultra-high-throughput microscopy have enabled a new generation of cell classification methodologies using image-based cell phenotypes alone. In contrast to current single-cell analysis techniques that rely solely on slow and costly genetic/epigenetic analysis, these image-based analyses allow morphological profiling and screening of thousands or even millions of single cells at a fraction of the cost, and have been proven to demonstrate the statistical significance required for understanding the role of cell heterogeneity in diverse biological applications, ranging from cancer screening to drug candidate identification/validation processes. This paper examines the efficacies and opportunities presented by machine learning algorithms in processing large scale datasets with millions of label-free cell images. An automatic single-cell classification framework using convolutional neural network (CNN) has been developed. A comparative analysis of its efficiency in classifying large datasets against conventional k -nearest neighbors (kNN) and support vector machine (SVM) based methods are also presented. Experiments have shown that our proposed framework can efficiently identify multiple types cells with over 99% accuracy based on the phenotypic label-free bright-field images; and CNN-based models perform well and relatively stable against data volume compared with kNN and SVM.

Index Terms—Cell classification, convolutional neural network, bright field imaging, multiclass classification.

I. INTRODUCTION

IMAGE-BASED cell analytic methodologies hold great promises to enable a new research paradigm in understanding cell heterogeneities and developments. With this class of image-based workflow, information about the underlying biological system is inferred solely by analyzing images of the target cells. Compared to conventional genetic and epigenetic-based workflows that are complex, time-consuming and costly, obtaining and analyzing cell images is a relatively simple and

Manuscript received February 8, 2018; revised July 14, 2018, August 20, 2018, September 6, 2018, and October 9, 2018; accepted October 22, 2018. Date of publication October 31, 2018; date of current version September 4, 2019. This work was supported by the Research Grants Council (RGC) of Hong Kong under Grants CRF C7047-16G, GRF 17245716, and GRF 17203217; and in part by the Croucher Innovation Award. (*Corresponding author: Hayden Kwok-Hay So.*)

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: nanmeng@eee.hku.hk; elam@eee.hku.hk; tsia@eee.hku.hk; hso@eee.hku.hk).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JBHI.2018.2878878

economical process. As a result, they promise unprecedented processing throughput and latency that make them attractive alternatives for tasks such as large-scale label-free rapid cell profiling and cell screening [1], [2].

However, existing microscopy technologies can either produce very high content images at low throughput, or they can image with high throughput by sacrificing image fidelity. As a result, to understand cell heterogeneity by using image-based methodologies remains a great challenge.

In this work, by exploiting recent advances in time-stretch microscopy technologies to produce high-content cell images with high throughput [3]–[6], we explore the opportunities and challenges in using modern deep learning algorithms for large-scale label-free multi-class cell classification. We hypothesize that the large quantity of high-content cell images will facilitate modern deep learning algorithms to achieve multi-class cell classification with accuracy and specificity comparable to conventional genetic and epigenetic methods. Starting with a convolutional neural network (CNN) classifier as a baseline, we compare its performance against a support vector machine (SVM) with the radial basis function (RBF) kernel and a k -nearest neighbor (kNN) classifier. We explain in detail a practical way to utilize CNN for learning useful features among different types of cells, focusing in particular on designing an appropriate framework by analyzing the learned features of each layer. The effect of data volume and image quality is also studied experimentally with the 3 classification methods.

Results show that the accuracy of our CNN classifier continues to increase with a larger volume of input data, approaching a peak classification accuracy of over 99%. On the other hand, while the classification accuracy of our SVM classifier initially benefits from a larger data volume, its performance is less stable and appears to suffer from overfitting, when the input training data volume increases to beyond 100 000. Finally, our kNN classifier, while relatively simple in design, fails to achieve comparable classification accuracy compared to the other 2 classifiers.

The rest of the paper is organized as follows. Section II reviews some previous work on cell classification. Section III presents our CNN-based framework and discusses the models and dataset. In Section IV, we report on the experimental investigation and comparisons. Section V details the experimental results and further analyzes the learned features to shed some light on the good performance of our proposed model. Finally, some concluding remarks are given in Section VI.

II. RELATED WORK

Many human somatic cell types can be highly dynamic in both morphology and behavior, even under homeostatic conditions. Researches encompassing technologies for characterizing heterogeneous behavior over time at the single cell resolution are of fundamental importance when studying rare events such as cancerization, or transition events such as differentiation. High-content screening (HCS) with high throughput biotechnology allows quantification of phenotypic changes of kinetic cells in various circumstances [7], but often depends on reliable fluorescent labels for accurate identification of cell characteristics [8]–[10]. The invasive and disruptive effects often associated with fluorescent dyes may limit the life-cycle and alter the characteristics of the cells under investigation [11], [12].

Over the years, many approaches have been proposed for cell classification involving fluorescent images, especially during the HEP-2 cell classification competitions. Previous attempts mostly consist of separate stages, where hand-crafted features are extracted and then fed into classifiers. Many popular image features, including intensity histogram, gray-level co-occurrence matrix, Fourier spectrum and wavelet coefficients, have been introduced to provide useful information for identification [13]. In addition, more elaborated features, such as the co-occurrence of adjacent local binary patterns (CoALBP), which can describe complex textures among adjacent patterns, have been utilized to learn the distribution of image micro-patterns as well [14], [15].

Recently, deep learning-based techniques have shown to outperform algorithms using hand-crafted features in many problems involving object classification and detection [16]–[24], and this has led to widespread interest in CNN-based classification model for cell recognition tasks [25], [26], such as mitosis detection in breast cancer histopathology images [27]. The benefits of deep learning architecture lead to the elimination of distinct feature extraction, as all the representations are learned directly from the training data. A recent work [28], using a model based on LeNet [29], demonstrates good performance and adaptability on HEP-2 cell classification. Another representative work of Liu *et al.* [30] provides a method to relieve the heavy reliance on a large dataset by combining deep neural network with autoencoder.

With more work to differentiate various cells based on the CNN model, deep learning has been increasingly accepted as a powerful technique for learning staining patterns from the fluorescent images. Compared with the time-consuming techniques to obtain such fluorescent labels, phenotypic methods have distinct advantages. High throughput imaging platforms have emerged to enable the study of cell biology across a large number of cell images [31], [32]. Bright field image yields the possibility of identifying more descriptive features such as cell shape and texture. A few recent publications have shown that incorporating data from bright-field microscopy can yield interesting results. Van Valen *et al.* [33] designed a deep CNN for quantitative analysis of single living cells, and the experiments present an improvement in semantic segmentation accuracy and a significant reduction in curation time. Cohen *et al.* [34] developed a computational method to predict cell division by

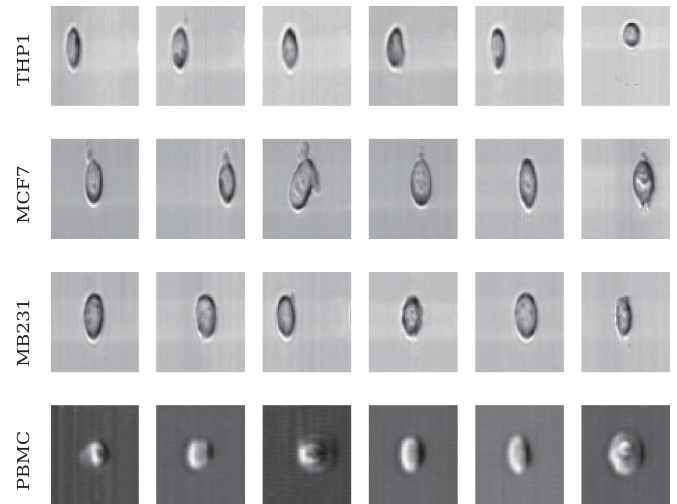


Fig. 1. Examples of single-cell images. Each row represents a specific type, from top to bottom: THP1, MCF7, MB231, and PBMC.

analyzing dynamic features of living cells. Meanwhile, Scherf *et al.* [35] in their work quantified changes in morphology of colonies of embryonic stem cells in different circumstances.

In this work, we propose a learning-based model for cell identification based on phenotypic images. Two major contributions distinguish our work from previous attempts. First, our model performs adaptively and noticeably well on large-scale multi-class cell dataset. It has been tested on the ultrafast imaging system for identifying four different types of cells, and experiments show that our system successfully recognizes them with over 99% accuracy. Moreover, all the scanning images are label-free without any staining patterns. Classification relies solely on image intensities, which is challenging due to cellular heterogeneity. Compared with two other real-time classification systems, CNN performs relatively stable along with data volume, while maintaining a high accuracy. Furthermore, we use t-distributed stochastic neighbor embedding (t-SNE) to show that the system is able to learn linearly separable features, and the probability map illustrates that those features are extracted mostly according to the cell edge textures.

III. METHOD

A. Data

We use an ultrafast imaging system known as asymmetric-detection time-stretch optical microscopy (ATOM) [3], [36]–[38] to collect a large dataset of cell images for training and evaluation. This unique imaging approach can achieve label-free and high-contrast flow imaging with good cellular resolution at a very high speed. In our dataset, each acquired image belongs to one of the four classes: THP1, MCF7, MB231 and PBMC, as shown in Fig. 1. More precisely, they belong to two collections of images captured separately at different instances. One contains over 8,000 samples of three types of cells (MCF7, PBMC, and THP1), while the other is significantly larger, with over 900,000 cell images of MB231, MCF7, and THP1 cells. Our objective is to augment the

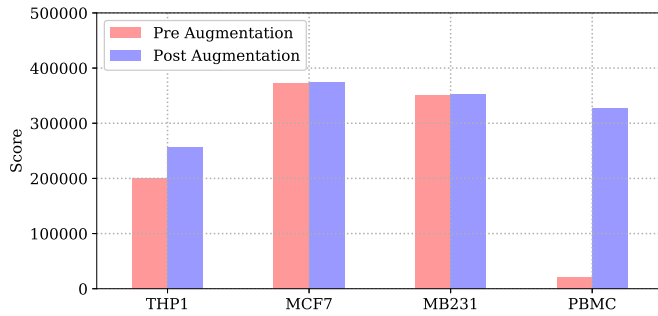


Fig. 2. The number of samples of different cell types.

imaging with a high-speed and robust classification system that is sensitive to the representations of different cell types.

The cell images are captured with different experimental settings such as illumination and focus, necessitating a classification system that is robust against these variations. Furthermore, the raw cell images are of different sizes, ranging from 21×102 to 68×440 pixels. Intentionally, some of the captured images also do not contain any cell sample. Thus, we first need to pre-process the images.

B. Image Pre-processing

Our first pre-processing step is to discard poorly formed source images, including those failing to capture a complete cell, containing defocus blur, or with some ghost artifacts. We use a simple classifier to select and abandon such images, noting that it is generally fine to be conservative and discard images whenever they appear to be problematic. For our dataset, this process drops 25.7% of the MB231 samples, 23.3% of MCF7, 37.5% of THP1, and a few samples of PBMC. Next, to unify the image dimension for subsequent training, we resize the samples to 128×128 . To deal with the smaller raw images, we upsample both horizontally and vertically using bilinear interpolation (better super-resolution scheme is also possible [39]); for others, we typically need to downsample the height while upsampling the width. We also normalize the average intensity to enhance the robustness to illumination fluctuations. Furthermore, considering that there are significantly fewer PBMC images than the other three types, we introduce data augmentation to balance the distribution. This is achieved through introducing affine transformations (flipping, translation, and rotation) on the existing cell images, as well as adding blur and noise to the image data. Fig. 2 shows the quantity of each type of cells before and after this process. This can reduce error due to skewed data.

C. Network Architecture

Image texture is a commonly used feature that can distinguish different types of cells efficiently [40]. The texture features can be described as a set of primitive pixels that provide the spatial information of image intensities in a region. To ensure information is not lost in the process, we keep a relatively large filter size (11×11) for the first convolutional layer, and gradually reduce the size of the filter to 3×3 . Image convolution highlights those local structures that resemble the convolution kernel throughout

TABLE I
NETWORK PARAMETERS

Layer number	Layer type	Hyper-parameter
Layer 1	Convolution	Filter size: 11×11 Feature bank: 32
Layer 2	Pooling	Pooling region size: 2×2 Pooling method: max-pooling Activation: ReLU
Layer 3	Convolution	Filter size: 6×6 Feature bank: 64
Layer 4	Pooling	Pooling region size: 2×2 Pooling method: max-pooling Activation: ReLU
Layer 5	Convolution	Filter size: 3×3 Feature bank: 128
Layer 6	Pooling	Pooling region size: 2×2 Pooling method: max-pooling Activation: ReLU
Layer 7	Full connection	Neurons: 64 Activation: ReLU
Layer 8	Full connection (softmax)	Neurons: 4 Activation: ReLU

an image region. In this way, the filters can learn texture-level features useful for the following fully connected layers.

On the basis of these analyses, our proposed network is designed as shown in Table I. Specifically, it contains eight layers. The first six layers are convolutional layers alternating with pooling layers, and the last two are fully connected layers for classification.

1) *Convolutional Layer*: For the k th convolutional layer, let $N^{(k)}$ denote the number of feature maps at this layer. Accordingly, each feature map is denoted as $\mathbf{h}_j^{(k)}$, where $j = 1, 2, \dots, N^{(k)}$. Each convolutional layer is parameterized by an array of two-dimensional filters $\mathbf{W}_{ij}^{(k)}$, which combine the i th feature map $\mathbf{h}_i^{(k-1)}$ in the previous layer with the j th feature map $\mathbf{h}_j^{(k)}$ in the present layer and the bias $b_j^{(k)}$. Each filter acts as a specific feature detector by convolving with the input feature map. In this way, all the filters pass through it with constant strides. The results are then fed into a nonlinear activation function $\psi(\cdot)$ such as the rectified linear unit (ReLU). Overall, the feature maps of the k th layer can be expressed as

$$\mathbf{h}_j^{(k)} = \psi \left(\sum_{i=1}^{N^{(k-1)}} \mathbf{h}_i^{(k-1)} * \mathbf{W}_{ij}^{(k)} + b_j^{(k)} \right), \quad (1)$$

for $j = 1, 2, \dots, N^{(k)}$, where $*$ represents the convolution operation.

2) *Pooling Layer*: A pooling layer performs the downsampling operation to the feature map. This greatly reduces the redundant information and also relieves the computation workload in training a CNN. Max-pooling and average-pooling are two commonly used techniques. The former picks the maximum pixel value as the representation of a small pooling region, while

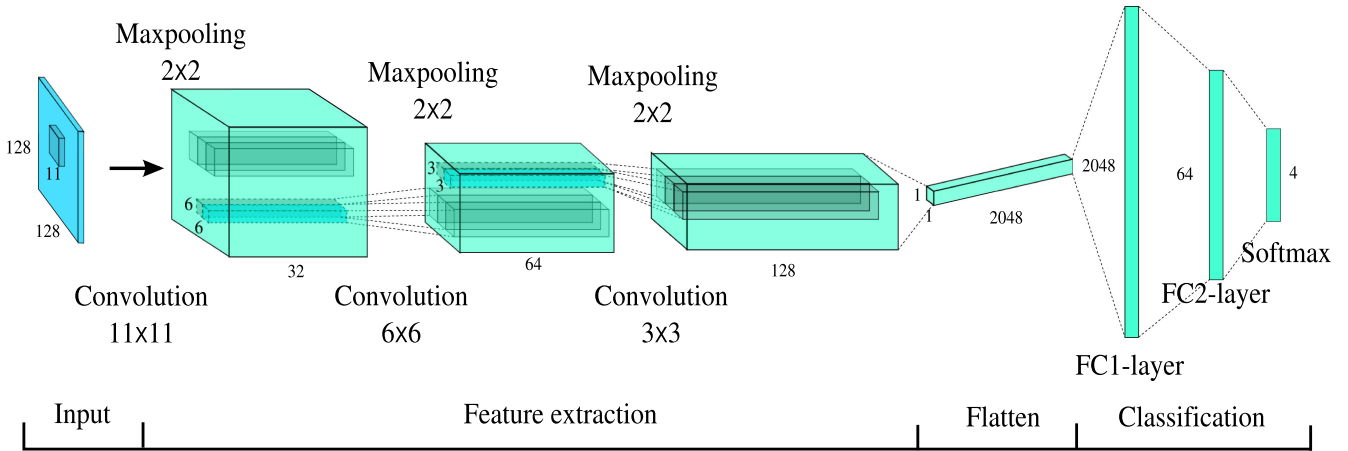


Fig. 3. The framework of our deep CNN classification system for cell images. Each block within the feature extraction stage denotes a feature map. The fully connected layer is abbreviated as FC. The annotations between two layers stand for the corresponding operation.

the latter uses the average. We make use of the former, which generally gives a better performance [41], [42].

3) *Classifier*: Classification layers are usually composed of one or more fully connected layers at the top of a CNN. Our network contains two, where the first one takes the output of Layer 6 as input, applies matrix multiplication with weights $\mathbf{W}^{(7)}$, and then sums with $\mathbf{b}^{(7)}$. The output of this layer is therefore

$$\mathbf{h}^{(7)} = \psi \left(\mathbf{W}^{(7)} \mathbf{h}^{(6)} + \mathbf{b}^{(7)} \right). \quad (2)$$

The second fully-connected layer is similar, and contains n neurons, which denotes the number of categories. The output probability $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ via softmax regression, where y_j is the output probability of the j th neuron, is

$$\mathbf{h}^{(8)} = \mathbf{W}^{(8)} \mathbf{h}^{(7)} + \mathbf{b}^{(8)}, \quad (3)$$

$$y_j = \frac{\exp \{h_j^{(8)}\}}{\sum_{i=1}^n \exp \{h_i^{(8)}\}}, \quad j = 1, 2, \dots, n. \quad (4)$$

Our network architecture is illustrated in Fig. 3. When a cell image is fed into the network, the spatial resolution of each feature map decreases as the features are extracted hierarchically from the earlier layer to the next one. These features are usually invariant to translational and small variations.

IV. EXPERIMENTS

A. Experimental Setup

The evaluation of our model is based on a train-validation-test scheme. To train our network, we randomly partition the over 1 million cell images into three subsets, with 65% for training (653,250 images), 15% for validation (150,750 images) and 20% for test (201,000 images). The training set allows the model to learn good representations, while the validation set is used for fine-tuning the hyper-parameters. The overall performance of the network is assessed on the test set. To give a comprehensive display of network performance, we use the F-score, which is

defined as the harmonic mean of precision and recall, and the average accuracy as evaluation metrics. Mathematically, let TP be the number of true positives, FP the number of false positives, and FN the number of false negatives. Precision and recall are given by

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (5)$$

and the F-score and average accuracy can be computed as

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (6)$$

$$\text{F-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (7)$$

B. Network Training

Training a network is essentially optimizing a nonlinear function with respect to weights and biases. The experiments are carried out using the Adam optimizer [43] to minimize the categorical cross entropy, which computes the dissimilarity of the approximated output distribution. The parameters include the learning rate, the exponential decay rate for the moving averages of the gradient, and the squared gradient. We set the final learning rate to be 0.0001, the decay rate to be 0.9, and the moving average to be 0.999. All the weights are initialized using truncated normalization, with a standard deviation of 0.1, and the biases are initialized with a random constant value. The weight updates are performed in every mini-batch, each containing 1300 images.

The network is implemented using TensorFlow [44] and computed on a Linux OS computer with an Intel Core i7-@2.60 GHz CPU and an Nvidia Tesla K40c card.

V. RESULTS

We evaluate our proposed CNN-based classification system on two datasets using the metrics defined in Eqs. (6)–(7). The experimental results are presented in three parts. First, we illustrate the cell images used for classification. Next, we explore

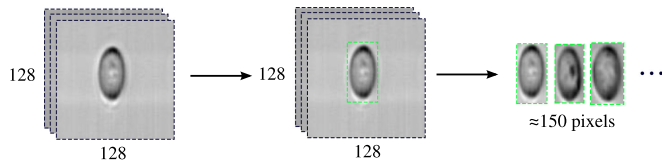


Fig. 4. The cell images used for training and evaluation of our model, together with the cell region enclosed by a tight bounding box.

the influence of the dataset size on the model performance, and compare with some traditional methods in order to provide an analysis of the differences between our work and some previous cell classification methods. Third, we give a comprehensive analysis of the performance of our network.

A sample cell image with size 128×128 is shown in Fig. 4. Note that the cell region only occupies around 150 pixels in total within a tight bounding box. All cell images used for training and evaluation are obtained directly from the ATOM imaging system with a high throughput [3]. To overcome the challenges in single-cell imaging, a large dataset can be very helpful, and therefore using this ultrafast imaging system is advantageous in allowing us to obtain enough cell images in a reasonable time.

A. Comparison of Methods

After applying the image pre-processing discussed earlier in our CNN model, the resulting images are fed to different classifiers for comparison. We implement these comparison methods where the parameters for each approach are picked using a trial-and-error procedure on the validation dataset. We compare mainly with two classical machine learning approaches, the first one being kNN, and the second one being SVM with the RBF kernel.

When compared to the CNN and SVM classification approaches where there are distinct training and inference phases, the kNN model classifies images by comparing the incoming image against previously encountered images. The incoming image is classified through a majority vote of its nearest neighbors in the training samples. In terms of computational resource requirements, kNN thus has a clear advantage over CNN and SVM when the amount of data volume is small. However, such advantage diminishes as the number of training samples increase. In our experiments, we classify images against a randomly subsampled set containing 6500 samples of our training data to limit its run time.

CNNs, on the other hand, can take advantage of increases in the amount of available computation and data. Indeed, our experiments demonstrate that the size of the dataset used for training-validation-test is crucial to tune it for better convergence. We first shuffle the image data and subsample it to obtain several smaller datasets. The subsampling rate ranges from 0.1 to 0.001, and we then test the performance of the three machine learning algorithms on each subset. In Fig. 5, their accuracies are compared across different data volumes. We observe that kNN is most insensitive to changes in data volumes. As the amount of data increases, learning-based techniques show greater potential. When the size of dataset is relatively small, the accuracy

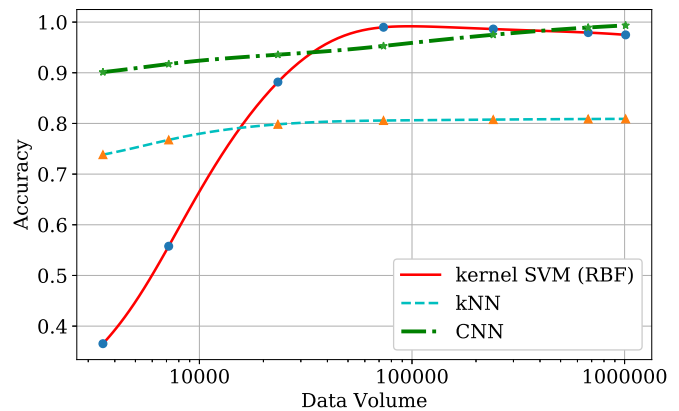


Fig. 5. Demonstration of the impact of data volume, comparing CNN with kNN and SVM with the RBF kernel.

curves of SVM increase rapidly as the amount of data increases. When the dataset becomes rather large, however, the performance becomes stagnant and the accuracy can even drop. This indicates that the additional data can hardly provide any additional useful knowledge in tuning the support vectors. In contrast, CNN delivers a good performance over all possible size of the dataset and exhibits its power for extracting representative features from the whole dataset, with accuracy steadily improving along with an increasing data size. Table II presents a more detailed view of the classification results using different methods, including a comparison with another recently-proposed CNN-based technique [28]. We adopted the CNN structure proposed by Gao *et al.* [28] to our cell classification problem. Our implementation strictly followed the framework presented in their paper except for changing the size of input images to 128×128 (the same input size of our model). In addition, the number of outputs in the final fully-connected layer is changed to 4 to meet the number of cell types in our dataset.

B. Analysis of the Network Performance

Although deep convolutional networks have brought breakthroughs in processing images, features automatically learned by a multilayer neural network are incomprehensible by humans most of the time. In this section, we provide some detailed discussion on the performance of our CNN-based model. Fig. 6 presents the validation accuracy and loss curve during the training process. Our model converges rapidly for the first 500 iterations, and continues to converge until around 3000 iterations. Both loss and accuracy curves are rather smooth with only small fluctuations. The validation loss continues to drop, reflecting that our proposed CNN-based model fits the problem well. Fig. 7 compares the model convergence of different loss functions. The model converges faster and better by using cross entropy based on softmax outputs as the loss function.

Next, we explore the learned features by probing into the filters of the convolutional layers. The features (filters) learned by the first and second layers after the training process are presented in Fig. 8. The filters of the first layer detect the texture features within a small region. The filters of the second layer

TABLE II
CLASSIFICATION ACCURACY OF KNN, SVM WITH RBF KERNEL, AND CNN WITH DIFFERENT DATA VOLUME

Methods		Data Volume						
		1100	6400	12000	64000	129000	651000	1305000
kNN	test	0.710	0.774	0.807	0.805	0.808	0.809	0.808
	validation	0.713	0.785	0.818	0.808	0.811	0.808	0.808
SVM	test	0.245	0.498	0.999	0.999	0.983	0.981	0.977
	validation	0.326	0.664	0.998	0.999	0.984	0.981	0.978
Gao et al. [28]	test	0.875	0.881	0.923	0.943	0.973	0.981	0.995
	validation	0.875	0.881	0.922	0.949	0.974	0.982	0.995
CNN	test	0.885	0.922	0.936	0.949	0.979	0.992	0.995
	validation	0.885	0.922	0.930	0.951	0.978	0.993	0.995

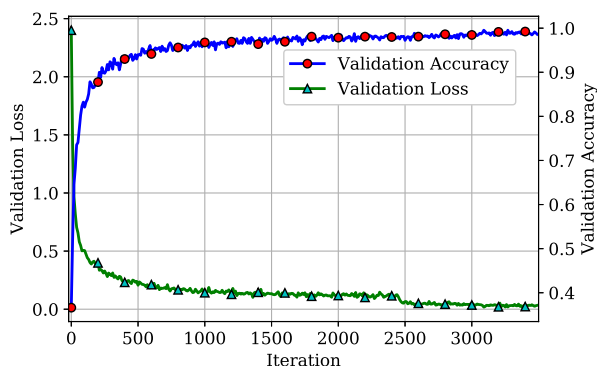


Fig. 6. The accuracy and loss curves of a batch of validation data during the training process.

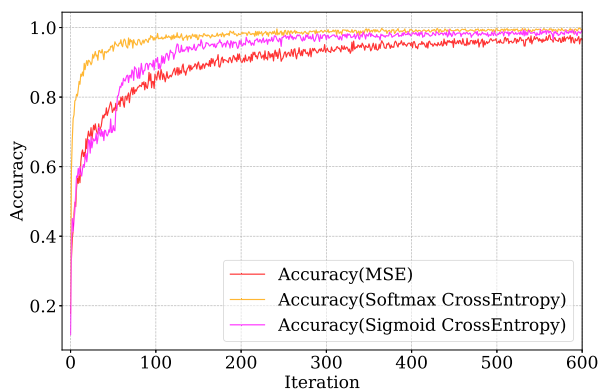


Fig. 7. The validation accuracy with respect to iteration during the training process. The cross entropy loss calculated based on softmax probability shows faster and better convergence when applied to training the proposed model.

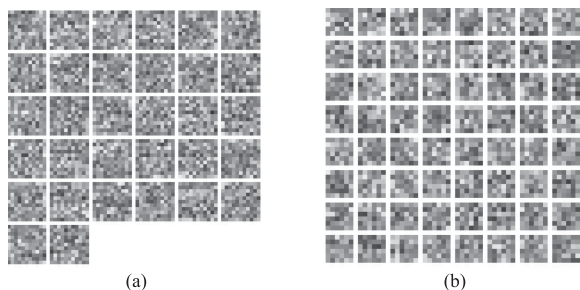


Fig. 8. Filters of the convolutional layers by training on our data. (a) The filters of the first layer. (b) The filters of the second layer.

TABLE III
COMPARISONS OF THE CLASSIFICATION PERFORMANCE ON THE VALIDATION DATASET AMONG KNN, SVM WITH RBF KERNEL, AND CNN. THE RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCE

Measurement	Methods	Cell Type			
		THP1	MCF7	MB231	PBMC
Precision	kNN	0.695	0.931	0.812	0.572
	SVM	0.999	0.999	0.999	0.485
	Gao et al. [28]	0.985	<u>0.996</u>	0.997	<u>0.991</u>
	CNN	<u>0.988</u>	<u>0.996</u>	0.999	0.995
Recall	kNN	0.827	0.782	0.829	0.697
	SVM	0.982	0.975	0.976	0.994
	Gao et al. [28]	<u>0.986</u>	<u>0.995</u>	<u>0.997</u>	<u>0.991</u>
	CNN	0.991	0.998	0.998	0.990
F score	kNN	0.755	0.850	0.820	0.628
	SVM	0.990	0.987	0.988	0.652
	Gao et al. [28]	0.986	<u>0.995</u>	<u>0.997</u>	<u>0.991</u>
	CNN	<u>0.989</u>	0.997	0.999	0.992

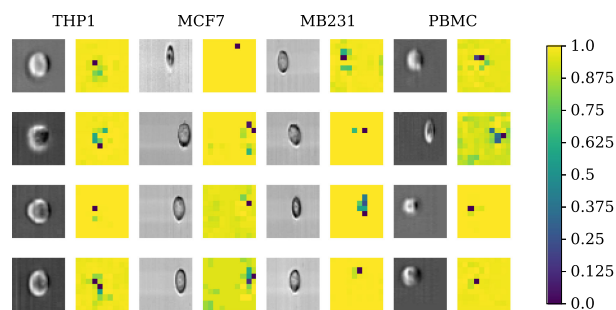


Fig. 9. The reverse probability hotmap to reflect the attention of our proposed network.

detect more elaborate textures based on the information of the first convolution layer.

We also use the t-SNE algorithm to visualize the learned representations after each layer. As shown in Fig. 10, a noticeable trend is that deeper network tends to present more distinguishable features. From the output of the first convolution layer to the second fully-connected layer, the point clouds of each cell types become progressively linearly separable. This reflects that our model can learn representative features of each class effectively. Table III and Table IV further provide a clearer and more

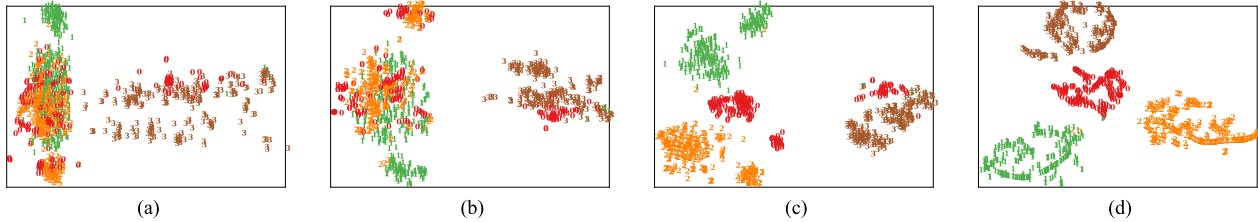


Fig. 10. The intermediate features extracted by the proposed network using t-SNE. We extract the intermediate output of each block and reduce the output dimension to two. (a) and (b) visualize the t-SNE of the second and third pooling layers. (c) and (d) show the results of the first and second fully connected layers. All the plots show only 1000 cell image samples randomly selected from the entire dataset.

TABLE IV

COMPARISONS OF THE CLASSIFICATION PERFORMANCE ON THE TEST DATASET AMONG KNN, SVM WITH RBF KERNEL, AND CNN. THE RED AND BLUE INDICATE THE BEST AND THE SECOND BEST PERFORMANCE

Measurement	Methods	Cell Type			
		THP1	MCF7	MB231	PBMC
Precision	kNN	0.700	0.929	0.811	0.587
	SVM	0.999	0.999	0.999	0.479
	Gao et al. [28]	0.987	0.995	<u>0.998</u>	<u>0.994</u>
	CNN	<u>0.989</u>	<u>0.996</u>	0.999	0.995
Recall	kNN	0.826	0.782	0.835	0.695
	SVM	0.982	0.974	0.976	<u>0.992</u>
	Gao et al. [28]	<u>0.990</u>	<u>0.997</u>	<u>0.998</u>	0.989
	CNN	0.992	0.998	0.999	0.991
F score	kNN	0.758	0.850	0.823	0.637
	SVM	0.991	0.986	0.988	0.646
	Gao et al. [28]	<u>0.989</u>	<u>0.996</u>	<u>0.998</u>	<u>0.992</u>
	CNN	0.991	0.997	0.999	0.993

comprehensive results of the classification performance on the validation and test datasets, showing that our CNN-based model is superior to the comparison methods.

Finally, Fig. 9 displays some randomly selected cell images and their corresponding reverse probability hotmaps. We use a small mask to cover each cell image and feed it into our proposed model. Then, we replace the covered region with the output probability value to obtain the hotmap. Since the covered region may contain some distinguishable features, the performance should deteriorate. In this way, we can visualize which part of the cell that CNN puts a larger weight. According to Fig. 9, cell types THP1 and PBMC contain some distinguishable features inside the cells, while for MCF7 and MB231, the cell edges have a heavier weight than inside.

VI. CONCLUSIONS

This paper proposes an automatic single-cell image classification framework with deep CNN. The network consists of 3 convolutional layers with 2×2 kernels and ReLU activations, followed by max-pooling layers with batch normalization operations. We give a detailed description on various aspects of this framework and discuss the data volume influence on classification performance, showing that deep learning models benefit more with a larger dataset. The proposed framework achieves

over 99% accuracy in identifying multiple types of cells with label-free images. In addition, the proposed CNN benefit most from the increased data volume when compared to the two other tested method with SVM and kNN.

REFERENCES

- [1] J. C. Caicedo *et al.*, "Data-analysis strategies for image-based cell profiling," *Nat. Methods*, vol. 14, no. 9, pp. 849–863, Sep. 2017.
- [2] M. Boutros, F. Heigwer, and C. Laufer, "Microscopy-based high-content screening," *Cell*, vol. 163, no. 6, pp. 1314–1325, Dec. 2015.
- [3] T. T. Wong *et al.*, "Asymmetric-detection time-stretch optical microscopy (ATOM) for ultrafast high-contrast cellular imaging in flow," *Sci. Rep.*, vol. 4, Jan. 2014, Art. no. 3656.
- [4] Q. T. K. Lai, K. C. M. Lee, A. H. L. Tang, K. K. Y. Wong, H. K.-H. So, and K. K. Tsia, "High-throughput time-stretch imaging flow cytometry for multi-class classification of phytoplankton," *Opt. Express*, vol. 24, no. 25, pp. 28 170–28 184, Dec. 2016.
- [5] A. C. Chan, H.-C. Ng, S. C. Bogaraju, H. K.-H. So, E. Y. Lam, and K. K. Tsia, "All-passive pixel super-resolution of time-stretch imaging," *Sci. Rep.*, vol. 7, Mar. 2017, Art. no. 44608.
- [6] J.-L. Wu *et al.*, "Ultrafast laser-scanning time-stretch imaging at visible wavelengths," *Light: Sci. Appl.*, vol. 6, no. 9, Jan. 2017, Art. no. e16196.
- [7] O. Z. Kraus and B. J. Frey, "Computer vision for high content screening," *Crit. Rev. Biochem. Mol. Biol.*, vol. 51, pp. 102–109, Jan. 2016.
- [8] O. Z. Kraus *et al.*, "Automated analysis of high-content microscopy data with deep learning," *Mol. Syst. Biol.*, vol. 13, Apr. 2017, Art. no. 924.
- [9] D. Fenistein, B. Lenseigne, T. Christophe, P. Brodin, and A. Genovesio, "A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening," *Cytometry Part A*, vol. 73A, pp. 958–964, Aug. 2008.
- [10] P. Arora *et al.*, "A rapid fluorescence-based assay for classification of iNKT cell activating glycolipids," *J. Amer. Chem. Soc.*, vol. 133, pp. 5198–5201, Mar. 2011.
- [11] M. Henriksen, B. Miller, J. Newmark, Y. Al-Kofahi, and E. Holden, "Laser scanning cytometry and its applications: A pioneering technology in the field of quantitative imaging cytometry," *Methods Cell Biol.*, vol. 102, pp. 159–205, 2011.
- [12] P. P. Laissue, R. A. Alghamdi, P. Tomancak, E. G. Reynaud, and H. Shroff, "Assessing phototoxicity in live fluorescence imaging," *Nat. Methods*, vol. 14, no. 7, pp. 657–661, 2017.
- [13] P. Soda and G. Iannello, "Aggregation of classifiers for staining pattern recognition in antinuclear autoantibodies analysis," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, pp. 322–329, Jan. 2009.
- [14] R. Nosaka, Y. Ohkawa, and K. Fukui, "Feature extraction based on co-occurrence of adjacent local binary patterns," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2011, pp. 82–91.
- [15] P. Foggia, G. Percannella, P. Soda, and M. Vento, "Benchmarking HEp-2 cells classification methods," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1878–1889, Jun. 2013.
- [16] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Euro. Conf. Comput. Vision*, 2014, pp. 818–833.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vision Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [18] S. Ren, K. He, R. Girshick, and S. Jian, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2016.

- [19] Z. Ren, Z. Xu, and E. Y. Lam, "Learning-based nonparametric autofocusing for digital holography," *Optica*, vol. 5, no. 4, pp. 337–344, Apr. 2018.
- [20] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017.
- [21] H. Lei *et al.*, "A deeply supervised residual network for HEP-2 cell classification via cross-modal transfer learning," *Pattern Recognit.*, vol. 79, pp. 290–302, Jul. 2018.
- [22] P. Eulenberg *et al.*, "Reconstructing cell cycle and disease progression using deep learning," *Nat. Commun.*, vol. 8, no. 1, 2017, Art. no. 463.
- [23] D. Pischel, J. H. Buchbinder, K. Sundmacher, I. N. Lavrik, and R. J. Flassig, "A guide to automated apoptosis detection: How to make sense of imaging flow cytometry data," *PLoS One*, vol. 13, no. 5, 2018, Art. no. e0197208.
- [24] Y. Jiang *et al.*, "Label-free detection of aggregated platelets in blood by machine-learning-aided optofluidic time-stretch microscopy," *Lab Chip*, vol. 17, no. 14, pp. 2426–2434, 2017.
- [25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. Comput. Vision Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
- [26] N. Meng, H. K.-H. So, and E. Y. Lam, "Computational single-cell classification using deep learning on bright-field and phase images," in *Proc. IAPR Conf. Mach. Vision Appl.*, May 2017, pp. 164–167.
- [27] M. Veta *et al.*, "Assessment of algorithms for mitosis detection in breast cancer histopathology images," *Med. Image Anal.*, vol. 20, no. 1, pp. 237–248, 2015.
- [28] Z. Gao, L. Wang, L. Zhou, and J. Zhang, "HEP-2 cell image classification with deep convolutional neural networks," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 2, pp. 416–428, Feb. 2017.
- [29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [30] J. Liu, B. Xu, L. Shen, J. Garibaldi, and G. Qiu, "HEP-2 cell classification based on a deep autoencoding-classification convolutional neural network," in *Proc. Int. Symp. Biomed. Imag.*, Jun. 2017, pp. 1019–1023.
- [31] O. Sandler, S. P. Mizrahi, N. Weiss, O. Agam, I. Simon, and N. Q. Balaban, "Lineage correlations of single cell division time as a probe of cell-cycle dynamics," *Nature*, vol. 519, pp. 468–471, Mar. 2015.
- [32] M. Wachsmuth *et al.*, "High-throughput fluorescence correlation spectroscopy enables analysis of proteome dynamics in living cells," *Nat. Biotechnol.*, vol. 33, pp. 384–389, Mar. 2015.
- [33] D. A. Van Valen *et al.*, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments," *PLoS Comput. Biol.*, vol. 12, Nov. 2016, Art. no. e1005177.
- [34] A. R. Cohen, F. L. A. F. G. Gomes, B. Roysam, and M. Cayouette, "Computational prediction of neural progenitor cell fates," *Nat. Methods*, vol. 7, no. 3, pp. 213–218, Feb. 2010.
- [35] N. Scherf *et al.*, "Imaging, quantification and visualization of spatio-temporal patterning in mESC colonies under different culture conditions," *Bioinformatics*, vol. 28, pp. i556–i561, Sep. 2012.
- [36] J. Xie, X. Niu, A. K. S. Lau, K. K. Tsia, and H. K.-H. So, "Accelerated cell imaging and classification on fpgas for quantitative-phase asymmetric-detection time-stretch optical microscopy," in *Proc. Int. Conf. Field Programmable Technol.*, Dec. 2015, pp. 1–8.
- [37] H. C. Ng *et al.*, "High-throughput cellular imaging with high-speed asymmetric-detection time-stretch optical microscopy under FPGA platform," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs*, Nov. 2016, pp. 1–6.
- [38] M. Wang *et al.*, "Real-time object detection and classification for high-speed asymmetric-detection time-stretch optical microscopy on FPGA," in *Proc. Int. Conf. Field-Programmable Technol.*, Dec. 2016, pp. 261–264.
- [39] R. Shi, A. C. Chan, E. Y. Lam, and H. K.-H. So, "Image super-resolution for ultrafast optical time-stretch imaging," in *Proc. Congr. Int. Commission Opt.*, Aug. 2017, Paper W1F–08.
- [40] K. Li, J. Yin, Z. Lu, X. Kong, R. Zhang, and W. Liu, "Multiclass boosting SVM using different texture features in HEP-2 cell staining pattern classification," in *Proc. Int. Conf. Pattern Recognit.*, Feb. 2012, pp. 170–173.
- [41] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2010, pp. 111–118.
- [42] Z. Ren, Z. Xu, and E. Y. Lam, "Autofocusing in digital holography using deep learning," in *Proc. SPIE Three-Dimensional Multidimensional Microsc.: Image Acquisition Process.*, Proc. Jan. 2018, p. 104991V.
- [43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [44] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operat. Syst. Des. Implementation*, 2016, pp. 265–283.