Title	A robust recognition error recovery for micro-flow cytometer by machine-learning enhanced single-frame super-resolution processing
Author(s)	Huang, Xiwei; Wang, Xiaolong; Yan, Mei; Yu, Hao
Citation	Huang, X., Wang, Xi., Yan, M., & Yu, H. (2014). A robust recognition error recovery for micro-flow cytometer by machine-learning enhanced single-frame super-resolution processing. Integration, the VLSI journal, 51, 208-218.
Date	2014
URL	http://hdl.handle.net/10220/24495
Rights	© 2014 Elsevier B.V. This is the author created version of a work that has been peer reviewed and accepted for publication by Integration, the VLSI Journal, Elsevier B.V. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [http://dx.doi.org/10.1016/j.vlsi.2014.07.004].

A Robust Recognition Error Recovery for Micro-flow Cytometer by Machine-learning Enhanced Single-frame Super-resolution Processing

Xiwei Huang, Xiaolong Wang, Mei Yan, Hao Yu*
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
Email: haoyu@ntu.edu.sg

Abstract:
With the recent advancement in microfluidics based lab-on-a-chip technology, lensless imaging system integrating microfluidic channel with CMOS image sensor has become a promising solution for the system minimization of flow cytometer. The design challenge for such an imaging-based micro-flow cytometer under poor resolution is how to recover cell recognition error under various flow rates. A microfluidic lensless imaging system is developed in this paper using extreme-learning-machine enhanced single-frame super-resolution processing, which can effectively recover the recognition error when increasing flow rate for throughput. As shown in the experiments, with mixed flowing HepG2 and Huh7 cells as inputs, the developed scheme shows that 23% better recognition accuracy can be achieved compared to the one without error recovery. Meanwhile, it also achieves an average of 98.5% resource saving compared to the previous multi-frame super-resolution processing.

Key words:
Microfluidic lensless imaging; Recognition error control; Single-frame super-resolution processing; Extreme learning machine.

1. Introduction

Flow cytometer has been widely deployed in biological research and clinical diagnostics to automatically determine the number or concentration of one or multiple types of cells [1-6]. The measurement is performed by passing a narrow stream of cells through a focused laser beam, and optical signals such as forward scattering (FSC), side scattering (SSC), fluorescent light emission (FL) are simultaneously measured to obtain cell information such as size, granularity or internal complexity [1-3]. It is, however, associated with high cost and low portability with the usage of bulky and expensive optic lens. Moreover, most existing flow cytometers rely on non-imaging techniques for cell recognition, hence are intrinsically less accurate than microscopy analysis. With the recent advancement in microfluidics based lab-on-a-chip technology, microfluidics-based lensless imaging system [7-14] provides a promising alternative to develop imaging-based micro-flow cytometer with system minimization.

The microfluidic lensless imaging system in [7-14] directly integrates microfluidic device on top of CMOS image sensor [14-15] in close proximity. The cells flow through a transparent poly-dimenthyl-siloxane (PDMS) microfluidic channel with spatially incoherent broadband white light illuminated from above. As such, the projected cell shadow images with classical diffraction pattern can be captured by the CMOS image sensor underneath without lenses as shown in Fig. 1(c-d). After processing the captured cell images, the cell type can be recognized, and the cell number can be enumerated.

Due to the limitation from the size of CMOS image sensor pixel, the lensless imaging system has limited resolution. When cells flow through the microfluidic channel with high speed, the motion blur or loss of resolution is further resulted with details of high spatial frequency filtered out. Thus, the directly projected cell shadow images are generally in low resolution (LR). The LR images have little information of cell internal structures and will result in recognition error, which is the primary challenge for developing imaging-based micro-flow cytometer when compared to the traditional flow cytometer. Therefore, one needs to develop recognition error recovery scheme for micro-flow cytometer.

In this article, a microfluidic lensless imaging system is introduced with recognition error recovery under various flow rates. It is based on an extreme-learning-machine [16] enhanced single-frame super-resolution (SR) processing [17-22]. By assuming randomly generated weights with single hidden layer, the Extreme Learning Machine (ELM) can provide fast solution for the generalized feed-forward neural networks with many applications such as bioinformatics, image processing, etc [16]. As shown in Fig. 1(d), the system is targeted for counting flowing cells that can achieve similar accuracy and throughput as the traditional flow cytometer. For resolution improvement, ELM based single-frame SR processing is employed, where static high-resolution (HR) cell images obtained from microscope are firstly classified and stored as off-line training cell image library, which contains the detailed internal cell structure information existing in high frequency (HF) components. Then, one model can be generated from the training library. Thus, one can deploy on-line single-frame SR processing to generate an SR image by recovering the necessary HF components for a LR cell image using the trained model. Note that the recovered SR image and HR image has the same size. Then the recognition of flowing cells can be thereby performed by checking for the strongest structure similarity of the SR image with the HR images of different types.

As in lensless microfluidic cytometer the cell motion blurs

or loss of resolution happen when increasing flow rate for throughput, here a robust recognition error recovery is developed by tuning the ELM based SR magnification factor under different flow rates. More specifically, if the flow rate is low, the SR magnification factor can be set low to meet the requirements of cell recognition accuracy. If the flow rate is high to cause more blur, the SR magnification factor is increased to recover cell images with more internal information for better recognition. Thus, the recognition error can be effectively controlled based on such a machine-learning enhanced SR processing. As shown in the experiments, with mixed flowing HepG2 and Huh7 cells as inputs, the proposed scheme shows 23% better recognition accuracy when compared to the one with no error recovery. Moreover, an average of 98.5% resource saving for single-frame SR can be achieved when compared to the previous multi-frame SR processing [8].

The rest of this paper is organized as follows. In Section 2, the related microfluidic systems are reviewed. Then, the proposed microfluidic lensless imaging system is introduced with recognition error recovery in Section 3. Next, the machine-learning enhanced SR processing is discussed in Section 4, and the detailed recognition error control flow is presented in Section 5. The simulation results are shown in Section 6 with conclusions in Section 7.

2. Related works

Fig. 1 shows some of the existing microfluidics-based lab-on-a-chip systems. Fig. 1(a) is a digital microfluidic system that uses electro-wetting-on-dielectric to manipulate discrete microfluidic droplets with an external camera for monitoring [23-25]. The error recovery is based on a cyberphysical resynthesis technique to recompute the droplet routing schedules [24]. The system depicted in Fig. 1(b) uses the lensless imaging principle for cell recognition and counting with static cells placed in between cover slides [10]. Different from the above mentioned, the microfluidic lensless imaging system [7-14] captures and processes the shadow images of flowing cells for recognition and counting. The main limitation comes from the limited resolution at high flow rate.

Resolution improvement is usually the first step before recognition, where SR processing can be utilized to produce HR images from observed LR images [17-22]. Previous microfluidic lensless imaging system [8] employs a multi-frame SR processing with one high resolution (HR) image reconstructed from a large number (40 to 100) of sub-pixel-shifted LR cell images. In order to accurately capture the sub-pixel motions within multiple frames, the cell detection uses a drop-and-flow scheme as shown in Fig. 1(c). It thereby requires a slow capillary flow speed that limits the throughput for the cell recognition and counting. In addition, the storage of multiple cell images is huge with hardware implementation. As such, a cost-effective single-frame SR processing is imperative to develop.

Moreover, to tackle the recognition error resulted from cell motion, active error control is needed, either from the microfluidic flow rate control to reduce the blur or increase the SR magnification factor to improve the resolution. Previous digital microfluidic system has already employed cyber-physical error recovery [23-26], but there is no work on recognition error recovery for the microfluidic lensless imaging system with flowing cells. As such, a quality-of-service (QoS) aware control is also required to consider recognition error recovery under various flow rates.

3. Microfluidic lensless imaging system with recognition error recovery

Towards accurate flowing cell recognition and counting, as shown in Fig. 2, the proposed microfluidic lensless imaging system with recognition error control includes three components discussed below, namely physical sensing, cyber recognition, and cyber-physical recognition error recovery. Table 1 summarizes the main notations used throughout this paper.

3.1. Physical sensing: microfluidic lensless imaging

In microfluidic lensless imaging system, the CMOS image sensor captures the projected LR cell shadow image $I_{m\times n}$, where m and n are row and column pixel numbers. Hence the spatial resolution R_{es} of $I_{m\times n}$ is determined by pixel size D_{pix}, i.e.,

$$R_{es} = D_{pix}. \tag{1}$$

In such a classical diffraction system, the shorter object distance D_{obj} is between the cell and sensor pixel array, the higher contrast C_{on} is for the captured LR cell image $I_{m\times n}$, i.e.,

$$C_{on} = f(D_{obj}) = A/\left(1 + \left(\frac{D_{obj}}{D}\right)^{B}\right) \tag{2}$$

where A is the contrast amplitude, D is the characteristic distance, and B is the shape parameter [27]. Thus, the microfluidic channel height H should be just slightly higher than the cell size. The channel width W is moderately selected to ensure that cells can flow smoothly in a straight and predictable manner [8].

The cell flow is driven by an external pump, which is programmable through its USB interface with external PC so that the flow rate R μL/min can be tuned in real time. Assume there is a single channel design on sensor surface, then the cell flow speed V becomes $V = R/(W \cdot H)$ considering the laminar flow effect.

To have high throughput detection, it requires the microfluidic flow rate R to be as high as possible. However, when the pump flow rate R is too high, the captured LR cell images will become blurred due to the shutter speed limitation from the CMOS image sensor. Assume the exposure time of the sensor is t_{exp}, then the pump flow rate R corresponding to p pixel blur is

$$R = W \cdot H \cdot V = W \cdot H \cdot p \cdot D_{pix}/t_{exp}. \quad (3)$$

The blurred LR cell images lose the characteristic to accurately recognize the cell.

3.2. Cyber recognition: ELM enhanced super-resolution

After obtaining the LR cell images $I_{m\times n}$ from the CMOS image sensor, the PC software processing starts to perform the SR processing and recognition.

We use an extreme-learning-machine [16] enhanced single-frame SR processing, namely ELM-SR, with details discussed in Section 4. Generally speaking, the ELM-SR is composed of off-line training and on-line testing. A series of t HR cell images $I_{M\times N}(HR)$ and their corresponding t LR cell images $I_{m\times n}$ are first selected as the off-line training library, where M and N are the row and column pixel numbers of HR images. Assume that the magnification factor between the LR and SR image is F_{mag}, then $M = m \cdot F_{mag}$ and $N = n \cdot F_{mag}$. After obtaining the training model that generates HF components from LR images, the on-line testing takes one input LR image $I_{m\times n}$ to provide one SR image $I_{M\times N}(SR)$ recovered.

The cell recognition is performed afterwards based on $I_{M\times N}(SR)$ by comparing it with the standard HR images $I_{M\times N}(HR)$ in the training library. For recognition of cell type, the similarity metric used here is the widely used *structural similarity* (SSIM) [28]. The SSIM is a full reference metric between 0 and 1 to indicate the similarity between one SR image with one distortion-free reference HR image by

$$SSIM(SR, HR) = \frac{(2\mu_{SR}\mu_{HR})(2\sigma_{SR,HR})}{(\mu_{SR}^2+\mu_{HR}^2)(\sigma_{SR}^2+\sigma_{HR}^2)}, \quad (4a)$$

where μ_{SR} and μ_{HR} are the means of the SR and HR images, σ_{SR}^2 and σ_{HR}^2 are the variances of the SR and HR images, and $\sigma_{SR,HR}$ is the covariance of the SR and HR images. It is proven to be consistent with human eye perception compared with traditional metric such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE) [28].

The mean SSIM ($MSSIM$) is the average of the SSIMs for one SR image with all the typical HR images,

$$MSSIM(SR, HRlib) = \frac{1}{t}\sum_{i=1}^{t} SSIM(SR, HR_i) \quad (4b)$$

where t is the number of training images in the HR training library. As such, the cell is classified to one type when it has the maximum $MSSIM$ with those typical HR cell images of the same type in the library. As one example illustrated in Fig. 4, the SR image shows larger *MSSIM* with cell type2 compared with type1 and type3, hence it is recognized as a type2 cell. Finally, all the recognized cells are differentiated and counted to obtain total counts or concentration ratio.

3.3. Cyber-physical recognition error recovery

However, cell recognition may have error when the flow rate is increased. The *MSSIMs* at low flow rate would be degraded, thus cannot be utilized to accurately differentiate the cell type at high flow rate. In that case, recognition error recovery becomes imperative. In this microfluidic lensless imaging system, the CMOS image sensor plays the role of physical sensing of LR image $I_{m\times n}$, and the control is realized through dynamic flow rate R and SR F_{mag} tuning based on the feedback from the cyber recognition.

Such a cyber-physical error control flow is described in the Algorithm 1 as follows. In the initialization step, based on one initial magnification factor F_{mag} and one flow rate R, an LR image $I_{m\times n}$ is input for ELM-SR processing. During recognition, $I_{m\times n}$ is recovered towards an SR image $I_{M\times N}$. Then the $MSSIM_i$ of different cell types can be calculated. By checking if the $MSSIM_i$ difference meets a predefined threshold, the cyber-physical controller tunes the flow rate R and F_{mag} accordingly for possible error recovery. When all cells are correctly recognized, the counting results can also be obtained.

Algorithm 1 Pseudo code for cyber-physical control

Initialization

1: Set flow rate R and SR magnification factor F_{mag}

2: Input LR cell image $I_{m\times n(i)}$

Recognition

3: Perform ELM-SR to LR $I_{m\times n(i)}$ and obtain SR $I_{M\times N(i)}$

4: Calculate $MSSIM_i$ with different cell types

5: Check if $MSSIM_i$ difference meets the predefined threshold

Recovery

6: Tune R and F_{mag} for the following input cells according to the $\Delta MSSIM_i$

Counting

7: Add cell *i* to the type with the maximum $MSSIM_i$

4. Machine-learning enhanced super-resolution processing

4.1. Super-resolution processing

The primary function for recognition error recovery in the proposed microfluidic lensless imaging system is based on the SR processing. Generally, SR algorithms can be categorized into single-frame and multi-frame based [21]. As the multi-frame SR reconstruction [8] needs to accurately control sub-pixel motions of each frame, it is not suitable for high-throughput.

The single-frame SR includes two classes, 1) interpolation and sharpening based [17]; and 2) example pattern based [19, 22]. The first one uses generic image interpolation techniques such as bilinear and bicubic interpolation, which cannot recover HF components from LR image, while the second one can extract new HF information. The machine learning based approach that explores the pattern of training images to recover SR images falls into the second class.

Among numerous machine learning algorithms [29-30], neural-network based one shows low complexity. Particularly, the ELM, which was developed for single-hidden-layer feed forward neural networks (SLFNs), has only one input layer, one hidden layer and one output layer as shown

in Fig. 5. It has a major merit of randomly generated weights between input layer and hidden layer, making it tuning-free without expensive iterative training process [16]. This advantage over other machine learning approaches such as support vector machine [29] and back propagation [30] makes it suitable for SR processing and recognition error recovery in microfluidic lensless imaging, since the number of training cell images can be large if one needs to cover all different cell types under different appearances.

4.2. ELM-based image super-resolution

In this paper, an ELM-based super-resolution processing for recognition error recovery is developed as described in Algorithm 2, which includes an off-line training and one on-line testing [19].

Algorithm 2 Pseudo code for ELM based super-resolution for recognition error recovery

ELM-SR Training:

1: Initialize a series of t LR inputs $I_{m\times n}$ and t HR inputs $I_{M\times N}(HR)$
2: Generate targeting HF row vector $\boldsymbol{T}$
3: Obtain t HF images $I_{M\times N}(HF) = I_{M\times N}(HR) - I_{M\times N}(LR\_Int\ )$
4: Generate row vector $\boldsymbol{T}$ based on t $I_{M\times N}(HF)$
5: Generate feature matrix $\boldsymbol{X}$
6: Bicubic interpolate t LR inputs $I_{m\times n}$ to $I_{M\times N}(LR\_Int\ )$
7: Extract pixel intensity distribution from $I_{M\times N}(LR\_Int\ )$
8: Generate the weight vector $\boldsymbol{\beta}$ with [$\boldsymbol{X}$, $\boldsymbol{T}$]
9: Generate rand matrix $\boldsymbol{A}$, $\boldsymbol{B}$, use sigmod function G
10: $\boldsymbol{T} = \boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}) = \boldsymbol{\beta}G(\boldsymbol{A}\boldsymbol{X} + \boldsymbol{B})$
11: $\boldsymbol{\beta} = \boldsymbol{T}\cdot\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}[\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{H}(\boldsymbol{X})\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}]^{-\boldsymbol{1}}$

ELM-SR Testing:

12: Input testing LR image $I_{m\times n}(LR')$
13: Generate feature matrix $\boldsymbol{X'}$
14: Bicubic interpolate $I_{m\times n}(LR')$ to $I_{M\times N}(LR\_Int')$
15: Extract pixel intensity distribution from $I_{M\times N}(LR\_Int'\ )$
16: Calculate HF components $\boldsymbol{I}_{M\times N}(HF')$

$$\boldsymbol{T}' = \boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}') = \boldsymbol{T}\cdot\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}\left[\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{H}(\boldsymbol{X})\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}\right]^{-\boldsymbol{1}}\boldsymbol{H}(\boldsymbol{X}')$$

17: Generate final SR output with HF components

$$\boldsymbol{I}_{M\times N}(SR) = \boldsymbol{I}_{M\times N}(LR\_Int'\ ) + \boldsymbol{I}_{M\times N}(HF')$$

4.2.1. ELM-SR training

As shown in the first part of Algorithm2, the off-line training aims to find a model to link the features in LR image with its HF features. The selected input features for ELM are the pixel intensity values, 1st order derivatives and 2nd order derivatives. The pixel intensity values and their distributions in HR cell image can represent the cell features. The 1st order derivatives in the horizontal and vertical directions can represent the directional change in pixel intensity values; and the 2nd order derivatives can further represent the rate of change in the 1st order derivatives. As such, one can perform the cell type recognition.

Firstly, t HR images $I_{M\times N}(HR)$ and their corresponding t LR images $I_{m\times n}(LR)$ are employed as the training library. To generate HF vectors, bicubic interpolation is used to obtain the interpolated images $I_{M\times N}(LR\_Int\ )$ from LR images $I_{m\times n}$. Then, HF images $I_{M\times N}(HF)$ are obtained by subtracting LR_Int from HR images, i.e.,

$$I_{M\times N}(HF) = I_{M\times N}(HR) - I_{M\times N}(LR\_Int\ ). \tag{5}$$

The t HF images $I_{M\times N}(HF)$ generate the targeting row vector $\boldsymbol{T}$ which contains the intensity values.

From line 5 to 7 of algorithm2, to extract the feature vector existed in $I_{M\times N}(\mathrm{LR\_Int}\ )$, a 3×3 pixel patch $P(i,j)$ centered at pixel (i,j) of $I_{M\times N}(\mathrm{LR\_Int}\ )$ is used to search through the whole $I_{M\times N}(\mathrm{LR\_Int}\ )$. Thus each patch generates a column vector containing 9 pixel intensity values, and four 1st and 2nd order derivatives in the horizontal and vertical directions, and one 2nd order mixed derivatives $(\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}, \frac{\partial^2 P}{\partial x}, \frac{\partial^2 P}{\partial y}, \frac{\partial^2 P}{\partial x \partial y})$. The whole t training images together with their column vectors form the feature matrix $\boldsymbol{X}$. Thus, the ELM training dataset $(\boldsymbol{X}, \boldsymbol{T})$ is generated.

Based on $(\boldsymbol{X}, \boldsymbol{T})$, the SR training can proceed to acquire $\boldsymbol{\beta}\epsilon R^L$, which is a row vector containing the weights between all the hidden nodes to the output node, as shown in line 8 to 11. In this ELM-SR model, there are d input nodes, L hidden nodes, and one output node. The output of the i^{th} hidden node is

$$h_i(x) = g(a_i \cdot x + b_i) = 1/(1 + \exp(-a_i \cdot x - b_i)) \tag{6}$$

where $a_i \epsilon R^d$ is a row vector containing the weights between all the input nodes to the i^{th} hidden node; b_i is a randomly generated bias term corresponding to the i^{th} hidden layer; g is the activation function of hidden layer, which we choose Sigmoid.

Then the output of the ELM is

$$f(\boldsymbol{x}) = \boldsymbol{\beta}\cdot\boldsymbol{h}(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{h}(x) = [h_1(x), h_2(x), \cdots, h_L(x)]^T$ is the output of the hidden layer. Then, the output matrix of hidden layer is

$$\boldsymbol{H}(\boldsymbol{X}) = G(\boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}), \tag{8}$$

where $\boldsymbol{A}$ is the weight matrix between input layer and hidden layer, $\boldsymbol{B}$ is the bias matrix, G is the same sigmoid function. Hence $\boldsymbol{T}$ is,

$$\boldsymbol{T} = \boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}). \tag{9}$$

In ELM-SR, both training error and the norm of output weights should be minimized, i.e.,

$$min\begin{cases}\|\boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}) - \boldsymbol{T}\| \\ \|\boldsymbol{\beta}\|\end{cases} \tag{10}$$

Then the orthogonal projection method can be used to acquire $\boldsymbol{\beta}$,

$$\boldsymbol{\beta} = \boldsymbol{T}\cdot\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}[\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{H}(\boldsymbol{X})\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}]^{-\boldsymbol{1}} \tag{11}$$

where $\boldsymbol{C}$ is a tuning parameter for the weight between $\|\boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}) - \boldsymbol{T}\|$ and $\|\boldsymbol{\beta}\|$, and $\boldsymbol{I}$ is the identity matrix with the same size to $\boldsymbol{H}(\boldsymbol{X})\boldsymbol{H}(\boldsymbol{X})^{\boldsymbol{T}}$.

4.2.2. ELM-SR testing

The second part of Algorithm 2 shows the ELM-SR testing. In the SR testing step, with an input LR image $I_{m\times n}(LR')$, we can easily predict the output using the same parameters $\boldsymbol{A}$, $\boldsymbol{B}$ and the acquired $\boldsymbol{\beta}$.

As line 13 to 15 of algorithm2 shows, the bicubic interpolation is still firstly performed to $I_{m\times n}(LR')$ to obtain $I_{M\times N}(LR\_Int')$. Then the same patch searching is applied to extract the feature matrix $\boldsymbol{X}'$. The output of the hidden layer is $\boldsymbol{H}(\boldsymbol{X}')$. As line 16 shows, based on $\boldsymbol{X}'$ and the already known parameters: $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\beta}$, we can obtain the output vector as

$$\boldsymbol{f}(\boldsymbol{X}') = \boldsymbol{\beta}\boldsymbol{H}(\boldsymbol{X}')$$

$$= \boldsymbol{T}\cdot\boldsymbol{H}(\boldsymbol{X})^{T}\left[\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{H}(\boldsymbol{X})\boldsymbol{H}(\boldsymbol{X})^{T}\right]^{-1}\boldsymbol{H}(\boldsymbol{X}') \quad (12)$$

Now $\boldsymbol{f}(\boldsymbol{X}')$ contains the estimated intensity values of HF components $I_{32\times 32}(HF')$. Finally, by extracting and adding the HF components to the interpolated image $I_{32\times 32}(\mathrm{LR\_Int}')$, the final SR image is recovered with sufficient internal details by

$$\boldsymbol{I}_{M\times N}(SR) = \boldsymbol{I}_{M\times N}(LR\_Int') + \boldsymbol{I}_{M\times N}(HF'). \quad (13)$$

4.2.3. Example for complexity analysis

To understand the implementation cost of ELM-SR, we take the example of 8×8 LR image as input, towards outputs of 32×32 (F_{mag}^{0}=4) and 72×72 (F_{mag}^{1}=9) HR images. In this ELM model, there are *d*=14 nodes in the input layer, *L*=20 nodes in the hidden layer, and one node in the output layer. For F_{mag}^{0}=4, each $I_{32\times 32}(LR\_Int)$ image contains 30×30=900 patches. With t training images, $\boldsymbol{X}$ contains 900t columns, and the HF intensity matrix $\boldsymbol{T}$ is a row vector with length 900t. HF image corresponds to 900 values in $\boldsymbol{T}$. Each value corresponds to one patch centre position.

In the on-line testing step with an input 8×8 LR' image, we firstly obtain $I_{32\times 32}(LR\_Int')$ from the input $I_{8\times 8}(LR')$. The feature matrix $\boldsymbol{X}'$ is extracted from LR_Int'. Certainly $\boldsymbol{X}'$ contains *d*=14 rows and 900 columns. The output of the hidden layer is $\boldsymbol{H}(\boldsymbol{X}')$. We can obtain the output based on (12). In this case, setting $\boldsymbol{C}$=512 can achieve a satisfied performance. The final SR image is then recovered by (13).

With $I_{8\times 8}$ LR input images, the number of addition and multiplication needed for super-resolution is counted for both F_{mag}^{0}=4 and F_{mag}^{1}=9 modes, and listed in Table 2 and Table 3, respectively. One can observe that the ratio of operation numbers between F_{mag}^{0}=4 mode and F_{mag}^{1}=9 mode is approximately 1:5.44, which growth is economical to be tuned in the cyber-physical control for recognition error recovery discussed below.

5. Cyber-physical control for recognition error recovery

5.1. Criterion for accurate cell recognition

As introduced in Section 3.2, cell recognition can be realized by comparing the recovered SR image with all the standard HR cell images of different types in the library and checking the *MSSIM* metric. As the cells of the same type have similar characteristics and hence have larger similarity, when all the *MSSIMs* for different types are obtained, the SR image can be classified into the cell type that has the biggest *MSSIM*. After each cell is recognized and classified into its type, the total cell number is accumulated and finally obtained. However, due to the possible blur effect of the captured LR image resulting from high flow rate, quality of the recovered SR image could be degraded, then the *MSSIMs* as well as the difference between them are also degraded.

Assume there are two types of cells, for each captured cell i with LR image $I_{m\times n(i)}$, its $MSSIM1_i$ and $MSSIM2_i$ corresponding to cell library 1 and 2 can be calculated as stated previously. To keep the recognition accuracy, the following criterion is checked on-line,

$$|MSSIM1_i - MSSIM2_i| > \Delta MSSIM. \quad (14)$$

where $\Delta MSSIM$ is a threshold *MSSIM* difference value. The criterion ensures that there is enough difference between the cell similarities with the two training cell types, thus the cell can be more accurately distinguished. This criterion can be achieved through the cyber-physical recognition error control discussed below.

5.2. Cyber-physical recognition error control

The cyber-physical recognition error control is shown in Algorithm 3. A series of cells are input with mixed types and sequences. In the initialization step, the flow rate controller sets one initial flow rate R^0 for cells in the first flow rate control cycle C_{tr} and the F_{mag} of the first cell to be F_{mag}^{0}. As flow rate R can introduce the cell blur effect, bigger F_{mag} makes the SR image showing more characteristics for the internal structure and vice versa. The on-line tuning of R and F_{mag} thereby can help with the recognition error control. As the tuning of physical pump flow rate needs certain responding time, the change of blur effect has some delay. Hence we define a flow rate control cycle C_{tr}.

Algorithm 3 Pseudo code for cyber-physical control of recognition error recovery

1: Initialize a series of Z LR inputs $I_{m\times n(i)}$
2: Initialize $R_i = R^0$ $(i \le C_{tr})$, $F_{mag(i)} = F_{mag}^{0}(i=1)$
3: for $1 \le i \le Z$
4: Perform ELM-SR to LR $I_{m\times n(i)}$ and obtain SR $I_{M\times N(i)}$
5: Calculate $MSSIM1_i$ and $MSSIM2_i$
6: if $|MSSIM1_i - MSSIM2_i| > \Delta MSSIM$
7: $R_{i+C_{tr}} = R_i + \Delta R1$
8: $F_{mag(i+1)} = F_{mag}^{0}$
9: else
10: $R_{i+C_{tr}} = R_i - \Delta R2$

11: $F_{mag(i+1)} = F_{mag}^{1}$
12: end
13: end

As shown in Algorithm 3, during the on-line recognition process, the cells are sequentially input for processing. Assume the current input cell is *i*, the captured LR image is $I_{m\times n(i)}$, and the ELM-SR is performed to recover the SR image $I_{M\times N(i)}$. Based on this image and the two HR training image libraries,

$$MSSIM1_i = \frac{\sum_1^{t_1} SSIM1_i}{t_1}$$

and

$$MSSIM2_i = \frac{\sum_1^{t_2} SSIM2_i}{t_2}$$

corresponding to two libraries can be obtained, where t_1 and t_2 are the numbers of training images of the two libraries.

We then check if criterion (14) can be met. If yes, indicating that the current SR image $I_{M\times N(i)}$ of cell *i* shows an obviously higher similarity to one of the cell libraries and it can be easily recognized. As such, the flow rate R can be further increased by $\Delta R1$ to improve the throughput, and F_{mag}^{0} is still kept for the recognition of next coming cell in the input sequence.

If criterion (14) cannot be met, indicating that current SR image $I_{M\times N(i)}$ of cell *i* has poor similarity to be easily distinguished between the two types. As such, the flow rate R need to be reduced by $\Delta R2$, and the big F_{mag}^{1} is applied for the next coming cell in the input sequence.

As a summary, with the defined flow rate control cycle C_{tr}, R can be changed according to

$$R_{i+C_{tr}} = R_i + \Delta R1 \text{ or } R_{i+C_{tr}} = R_i - \Delta R2.$$

While the online SR recovery is by software processing that reacts in real-time, which tunes the F_{mag}according to

$$F_{mag(i+1)} = F_{mag}^{0} \text{ or } F_{mag(i+1)} = F_{mag}^{1}.$$

Fig. 6 depicts an example with $C_{tr} = 5$. Note that we set $\Delta R1$ to be smaller than $\Delta R2$ so that flow rate and cell blur increases gradually for higher throughput.

5.3. System flow chart

The whole system processing flow is summarized in Fig. 7. The four steps in dashed line box are offline ELM-SR training, where $I_{m\times n}(LR)$ and $I_{M\times N}(HR)$ cell images are the inputs and they will generate $I_{M\times N}(LR\_int)$ and $I_{M\times N}(HF)$images as Section 4.2.1 discussed. With the off-line training model obtained, the on-line ELM-SR processing and recognition can be done as Section 4.2.2 and 5 discussed. The cell counting is conducted to accumulate each cell after its type is recognized. With the above mentioned cyber-physical recognition error control scheme, the microfluidic lensless imaging system can reach high-throughput detection with controlled accuracy.

6. Experimental Results

Different types of cell images such as bone marrow stromal cell and tumor cells (Hepg2 and Huh7) are used for experiments.

6.1. Flow Rate and Blur

Assume a CMOS image sensor of $D_{pix} = 3.2\mu\text{m}$ is used with microfluidic channel W=100μm and H=30μm in the microfluidic lensless imaging system. Fig. 8. shows an original 32×32 HR bone marrow stromal cell image taken by one Olympus IX71 microscope at X40 objective and its corresponding 8×8 lensless LR images of various blur effects. From the HR cell image we can see the dark cell boundary with light and shade internal structures, which are lost in the LR images and worse for blurred ones. With t_{exp}=0.1ms, we can obtain from (3) that 2 pixel, 3 pixel, and 4 pixel blur corresponding to the pump flow rates of 7.92μL/min, 11.88μL/min, and 15.84μL/min. The quality of the captured LR cell image is poor with larger blur when increasing the flow rate.

6.2. Single-frame vs. Multi-frame SR

Previous microfluidic lensless imaging system applies multi-frame SR processing to improve the resolution of LR cell images [8] that requires huge data to process. To compare the resource utilization with single-frame SR, we implement both SR processing algorithms by Xilinx Virtex-6 XC6VLX240T FPGA [32].

Table 4 shows the FPGA synthesis results of F_{mag}=4 for image size of 256×256 with one HR reconstructed from 5 LR images. The image storage for multi-frame SR is implemented by 294 block-RAMs, and for single-frame SR only registers are enough for row data storage. The block-RAM resource scales linearly with the number of LR frames. An average of 98.5% resource saving from 18234 slice registers, 154468 slice LUTs to only 469 registers and 672 LUTs can be seen for the single-frame SR processing compared to the multi-frame SR processing.

6.3. Performance of ELM-SR

The ELM-SR processing is implemented with MATLAB R2010b (MathWorks, MA) and run on a laptop computer (2.53-GHz Intel Core i5). We show the performance of the ELM-SR by improving the LR images of HepG2 and Huh7 cells.

6.3.1. Generation of training image library

As shown by the flow chart in Fig. 7, an off-line training image library of HR HepG2 and Huh7 should be built before performing the cell recognition. The raw HR images of HepG2 and Huh7 cells are taken by the Olympus IX71 microscope at X40 objective. And we save them into the HR image library with the size of 32×32 and 72×72.

During the HR image capturing phase, the corresponding 8×8 LR cell images are captured by the microfluidic lensless imaging system. Next, these LR cell images are interpolated to the same size of HR images, i.e., 32×32 and 72×72. Note that the detailed structures cannot be observed because the interpolation cannot recover the HF components. Then the HF components for each training cell image are obtained by subtracting the 32×32 interpolated cell images from the original 32×32 HR images. The same is done for the 72×72 images. After all these steps, the training library is generated, and it is input to the ELM-SR training model. Fig. 9 shows some of the training images.

Note that this paper is towards automatic cell counting such that we assume that the cell types are all known or to be pre-characterized in advance. Moreover, the extensive off-line training image library is generated in experiment for cell types of HepG2 and Huh7 under different rotation conditions. If a new pattern appears, we need to train a new ELM-SR model, which is off-line.

6.3.2. Characterization for number of ELM-SR training images

As shown by the flow chart in Fig. 7, the next step is to perform the ELM-SR training based on the obtained training image library.

Intuitively, the more training images used in one library, the better training model one can generate, which can recover SR images of higher SSIM with original HR images. To evaluate this trend and find a threshold training image number, we choose different numbers of training images, ranging from 2 to 27, to train different models, and use three different LR cells for testing. As shown from Fig. 10, we found that after using 18 training images, the recovered SR image almost keep the same *MSSIM* level with its original image. Hence, in the following system simulation, 27 training images are selected for each type of cell.

6.3.3. Effect of magnification factor

As shown by the flow chart in Fig. 7, the next step is to apply the trained model for ELM-SR testing. Since the F_{mag} will be tuned, effects of different F_{mag} are examined. Based on LR 8×8 HepG2 and Huh7 cell images, we perform the ELM-SR processing with F_{mag}^0=4 and F_{mag}^1=9. The 32×32 and 72×72 SR images are obtained as shown in Fig. 11. It is clearly perceived from the four SR images (a3, a4, b3, b4) that ELM-SR can recover the internal structures from LR image. Moreover, by increasing the magnification factor, the characteristic of original HR image can be revealed more. Such as in the 72×72 HepG2 SR image (a4), the white circle inside the black cell border is recovered, which is not shown in 32×32 Hepg2 SR image (a3), making it easier to be recognized and classified. The *MSSIMs* for 32×32 and 72×72 HepG2 and Huh7 cell images are 0.5010 and 0.5221, 0.3354 and 0.3516, respectively.

We also examine the effects of F_{mag} at different blur effects, which are characterized by the blurred pixel numbers. Take one HepG2 cell as the input, and use F_{mag}^0=4 and F_{mag}^1=9 to recover an SR image. Then, it is compared with both HepG2 and Huh7 libraries to check *MSSIMs*. As shown in Fig. 12, all *MSSIMs* reduce with the blur, as blur reduces the LR image quality compared to the images in both libraries. It is also noticeable that by using larger F_{mag}^1=9, the *MSSIM* difference between two types are about 40% larger than that obtained by F_{mag}^0 =4, which indicates easier and more accurate recognition. When pixel blur increases, The *MSSIM* difference reduces faster for small F_{mag}^0=4, but still remains at relative constant level for F_{mag}^1=9. This demonstrates the effectiveness of increasing the F_{mag} in recognition with increased flow rates.

6.4. Recognition error control

As shown by the flow chart in Fig. 7, the next is to apply the ELM-SR for cyber-physical recognition error control. A series of 100 captured LR HepG2 cell images and Huh7 cell images without blur effect are sequentially used as the simulation inputs with the sequence of each type known in advance.

To show the effectiveness of recognition error control, we check the recognition results with and without control. From the analysis of *MSSIM* at different blur effects for a number of input LR cell images, we empirically choose the control threshold *ΔMSSIM*=0.13, flow rate increase step *ΔR1*=0.2 and flow rate decrease step *ΔR2*=0.3. The two flow-rate tuning steps are directly set corresponding to the number of blurred pixels. The decrease-step is a bit larger than the increase-step such that when recognition error happens due to fast flowing speed and large blur, the flow rate and hence blur can be rapidly reduced. The magnification factor used, the blur effect and the recognition errors are shown in Fig. 13, where Error=0 indicates the cell is correctly recognized, and Error=1 means that the cell recognition is incorrect. It can be observed that with the flow rate and magnification factor control, the lensless microfluidic imaging system can reach a recognition accuracy without error. For the computation used, 41% of the total input cell images apply magnification factor F_{mag}^1=9 for ELM-SR processing, and the other 59% use magnification factor F_{mag}^0=4.

When there is no flow rate control, we input the cells as the same flow rate with the same blur effect of 4 pixels, then check the recognition accuracy and the computation resources used. The results are shown in Fig. 14. Now, the recognition accuracy only reaches 77% with recognition errors occur. Seen from the magnification data, the computation used is 58% of F_{mag}^1=9 and 42% of F_{mag}^0=4.

From the example above we can see that the cyber-physical recognition error control can ensure 23% better recognition accuracy. Meanwhile, as discussed in Section 4.2.3, the ratio of operation numbers between $\mathrm{F}_{\mathrm{mag}}^0$=4 mode and $\mathrm{F}_{\mathrm{mag}}^1$=9 mode is approximately 1:5.44. As such, the computation ratio with and without error control is

(0.41×5.44+0.59):(0.58×5.44+0.42) = 2.82:3.57. Therefore, the magnification factor tuning helps save the total computation by 21% due to less use of the computation intensive magnification factor F_{mag}^{1}=9 for ELM-SR processing.

7. Conclusions

One recognition error recovered lensless microfluidic imaging system is proposed for high-throughput cell recognition and counting. It is based on an extreme learning machine enhanced single-frame super-resolution processing. The recognition error control enables an on-line super-resolution magnification factor tuning for various flow rates. With mixed flowing HepG2 and Huh7 cells as inputs, simulation results show that 23% better recognition accuracy is achieved with the proposed scheme when compared to the one with no error recovery. It also achieves an average of 98.5% resource saving when compared to the previous multi-frame super-resolution processing. The proposed system has thereby provided a promising lab-on-a-chip design for the future realization of micro-flow cytometer.

References

[1] H. M. Shapiro, Practical flow cytometry, John Wiley & Sons, 2005.

[2] A. Givan, Flow cytometry: first principles, John Wiley & Sons, 2004.

[3] M. Brown, C. Wittwer, Flow cytometry: principles and clinical applications in hematology, Clin. Chem. 46 (2000) 1221-1229.

[4] G. P. Gupta, J. Massagué, Cancer metastasis: building a framework, Cell 127 (2006) 679-695.

[5] S. M. Hammer, J. J. Eron, P. Reiss, R. T. Schooley, M. A. Thompson, et al., Antiretroviral treatment of adult HIV infection, J. American Medical Association 300 (2008) 555-570.

[6] B. Nicole, M. Roederer, A practical approach to multicolor flow cytometry for immunophenotyping, J. Immunol. Methods 243 (2000) 77-79.

[7] X. Heng, D. Erickson, L. R. Baugh, Z. Yaqoob, P. W. Sternberg, D. Psaltis, C.-H. Yang, Optofluidic microscopy—a method for implementing a high resolution optical microscope on a chip, Lab Chip 10 (2006) 1274-1276.

[8] G. Zheng, S. A, Lee, S. Yang, C.-H. Yang, Sub-pixel resolving optofluidic microscope for on-chip cell imaging, Lab Chip 10 (2010) 3125-3129.

[9] S. A Lee, R. Leitao, G. Zheng, S. Yang, A. Rodriguez, C.-H. Yang, Color capable sub-pixel resolving optofluidic microscope and its application to blood cell imaging for malaria diagnosis, PLOS ONE 6 (2011): e26127.

[10] A. Ozcan, U. Demirci, Ultra wide-field lens-free monitoring of cells on-chip, Lab Chip 1 (2008) 98-106.

[11] T.-W. Su, S. Seo, A. Erlinger, A. Ozcan, High-throughput lensfree imaging and characterization of a heterogeneous cell solution on a chip, Biotechnol. Bioeng. 102 (2009) 856-868.

[12] Q. Wei, E. McLeod, H. Qi, Z. Wan, R. Sun, A. Ozcan, On-Chip cytometry using plasmonic nanoparticle enhanced lensfree holography, Scientific Reports 3 (2013) 1699.

[13] S. Moon, H. O. Keles, A. Ozcan, A. Khademhosseini, E. Hæggstrom, D. Kuritzkes, U. Demirci, Integrating microfluidics and lensless imaging for point-of-care testing, Biotechnol. Bioeng. 24 (2009) 3208-3214.

[14] M. Yan, X. Huang, Q. Jia, R. Nadipalli, Y. Shang, H. Yu, M. Je, K. S. Yeo, High-speed CMOS image sensor for high-throughput lensless microfluidic imaging system with point-of-care application, in: Proc. SPIE Electronic Imaging 2012, pp. 829804.

[15] A. J. P. Theuwissen, CMOS image sensors: state-of-the-art, Solid-State Electronics 52 (2008) 1401-1406.

[16] G. B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Systems, Man, and Cybernetics 42 (2012) 513-29.

[17] T. Wang, X. Huang, Q. Jia, M. Yan, H. Yu, K. S. Yeo, A super-resolution CMOS imager for microfluidic imaging applications, in: Proc. IEEE Biomedical Circuits and Systems Conf., 2012, pp. 388-391 .

[18] E. Choi, J. Choi, M. G. Kang, Super-resolution approach to overcome physical limitations of imaging sensors: an overview, Int. J. Imaging and System Technol. 14 (2006) 36-46.

[19] L. An, B. Bhanu, Image super-resolution by extreme learning machine, in: Proc. IEEE Int. Conf. on Image Processing, 2012, pp. 2209-2212.

[20] Z. Lin, J. He, X. Tang, C.-K. Tang, Limits of learning-based superresolution algorithms, in: Proc. IEEE Int. Conf. Computer Vision, 2007, pp. 1-8.

[21] S. C. Park, M. K. Park, M. G. Kang, Super-resolution image reconstruction: a technical overview, IEEE Signal Processing Magazine 20 (2003) 21-36.

[22] W. T. Freeman, E. C. Pasztor, Learning low-level vision, in: Proc. IEEE Int. Conf. Computer Vision, 1999, pp. 1182-1189.

[23] T.-W. Huang, T.-Y. Ho, K. Chakrabarty, Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips, in: Proc. IEEE/ACM Int. Conf. Computer-Aided Design, 2011, pp. 448-455.

[24] Y. Luo, K. Chakrabarty, T.-Y. Ho, Error recovery in cyberphysical digital-microfluidic biochips, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 32 (2013) 59-72.

[25] Y.-L. Hsieh, T.-Y. Ho, K. Chakrabarty, Design methodology for sample preparation on digital microfluidic biochips, in: Proc. IEEE Int. Conf. Computer Design, 2012, pp. 189-194.

[26] H. Qian, X. Huang, H. Yu, C.-H. Chang, Cyber-physical thermal management of 3D multi-core cache-processor system with microfluidic cooling, ASP J. Low Power Electronics 7 (2011) 110-121.

[27] H. Ji, D. Sander, A. Haas, P. A. Abshire, Contact imaging: simulation and experiment, IEEE Trans. Circuits and Systems I 54 (2007) 1698-1710.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Processing 13 (2004) 600-612.

[29] C. Cortes, and V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273-297.

[30] B. Yegnanarayana, Artificial neural networks, Prentice-Hall of India Pvt.Ltd, 2004.

[31] Xilinx Virtex-6 XC6VLX240T FPGA (2013, December). [On-line] Available: http://www.xilinx.com/products/boards-and-kits/EK-V6-ML605-G.htm

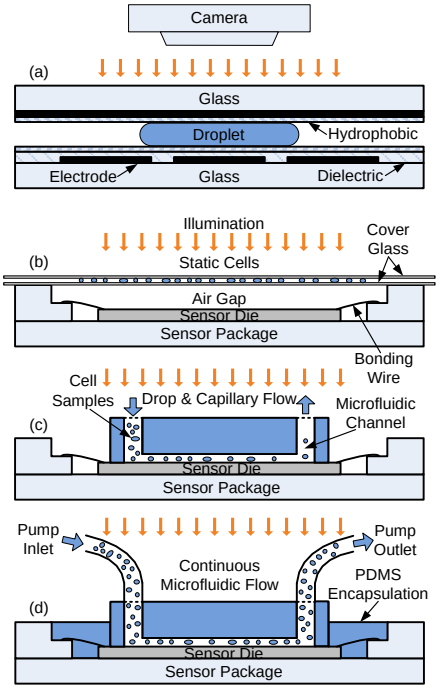

Fig.1. (a) Digital microfluidic system, (b) Static lensless imaging system, (c) Microfluidic lensless imaging system with capillary flow, (d) The proposed microfluidic lensless imaging system with continuous flowing cell detection.

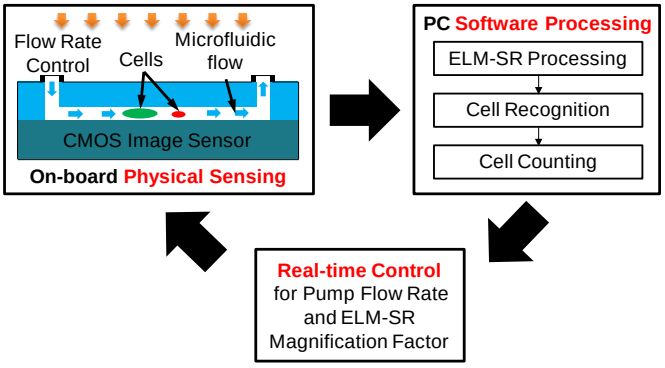

Fig. 2. Microfluidic lensless imaging system with recognition error control.

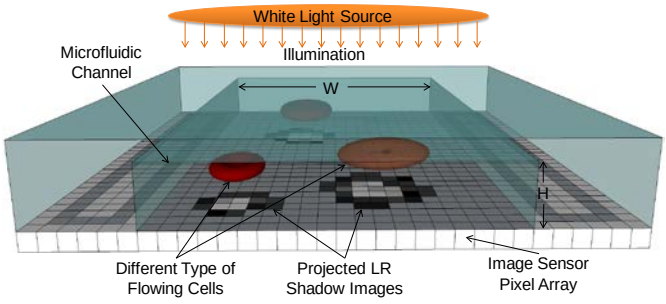

Fig. 3. Cell shadow image by microfluidic lensless imaging.

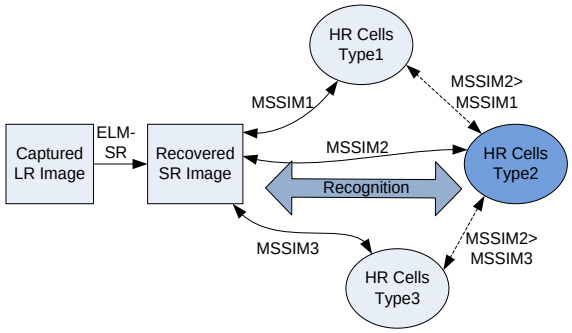

Fig. 4. The illustration of cell recognition based on the comparison of MSSIM.

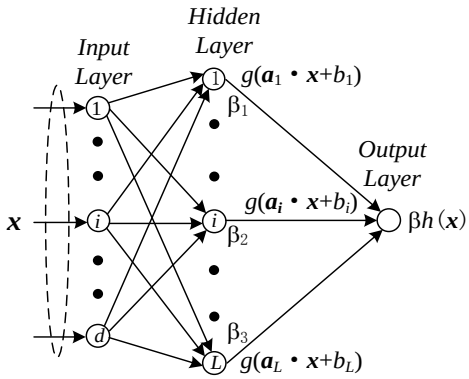

Fig. 5. Structure of the extreme learning machine model.

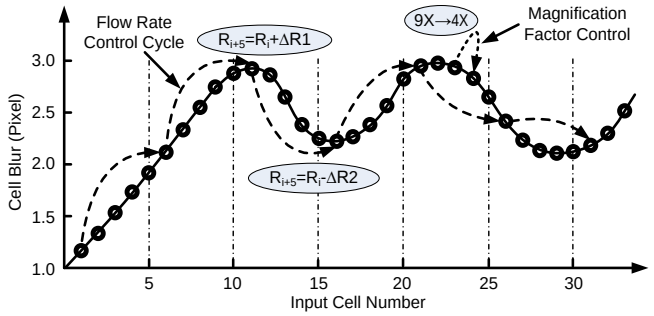

Fig. 6. Flow rate and magnification factor control for the recognition error recovery.

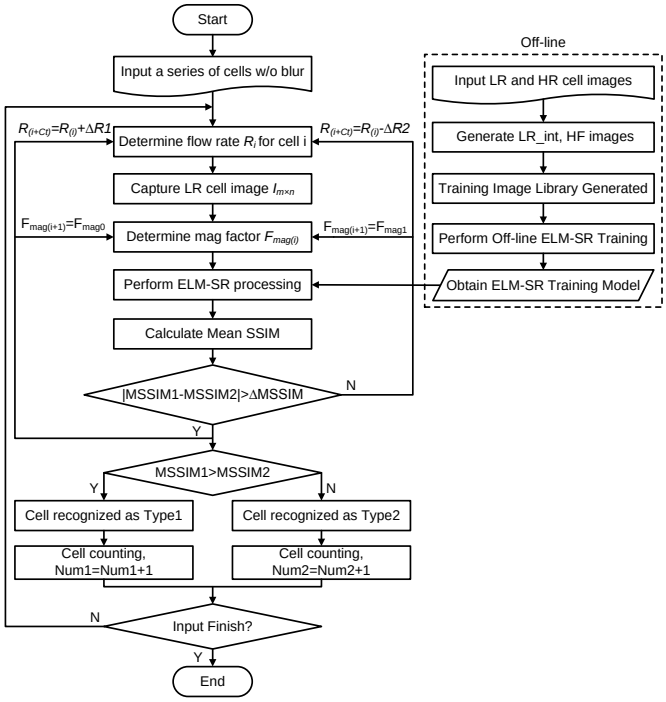

Fig. 7. The overall flow chart of the proposed microfluidic lensless imaging system with cyber-physical control of recognition error recovery.

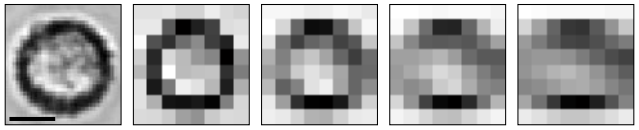

Fig. 8. The original HR cell image and its corresponding LR images with different blur effects under various flow rates. The scale bar is 10μm.

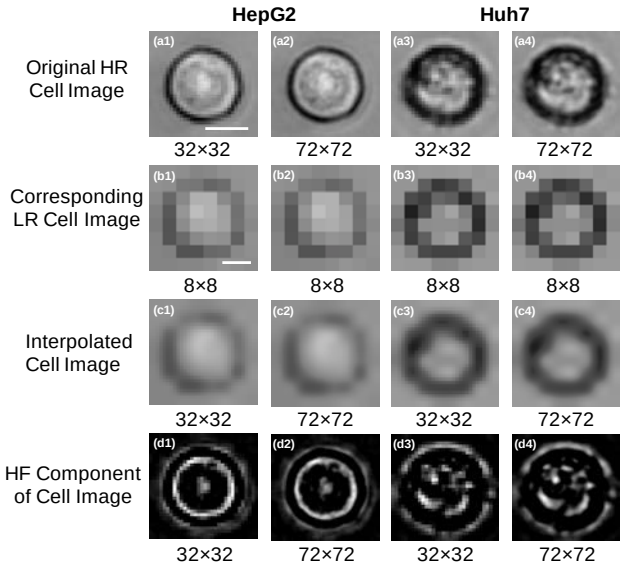

Fig. 9. Some image examples for the training image library, including original HR images, the corresponding LR cell images, interpolated images and the HF images.

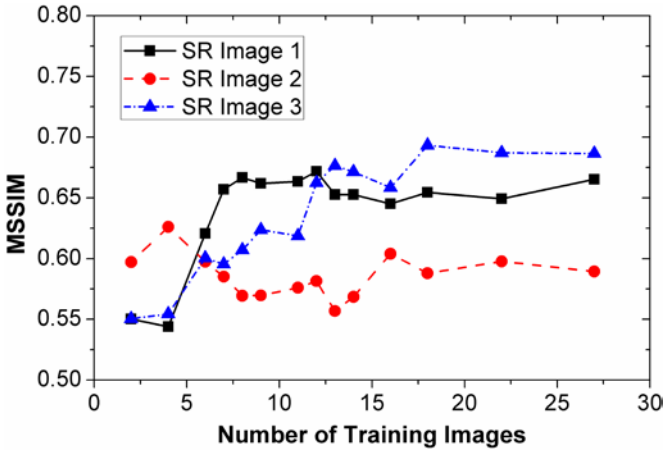

Fig. 10. The evolution of MSSIM with respect to the number of training images used.

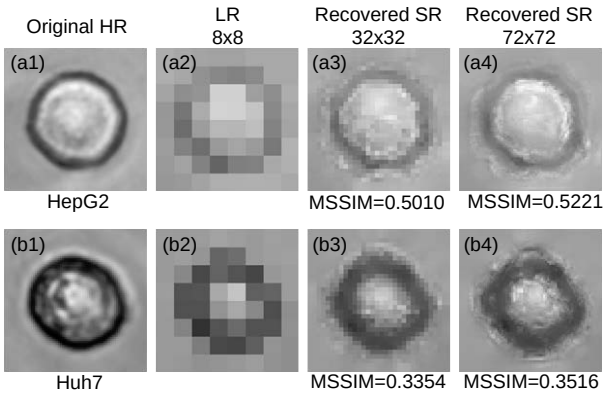

Fig. 11. The ELM-SR performance on recovering the LR HepG2 and Huh7 cell images.

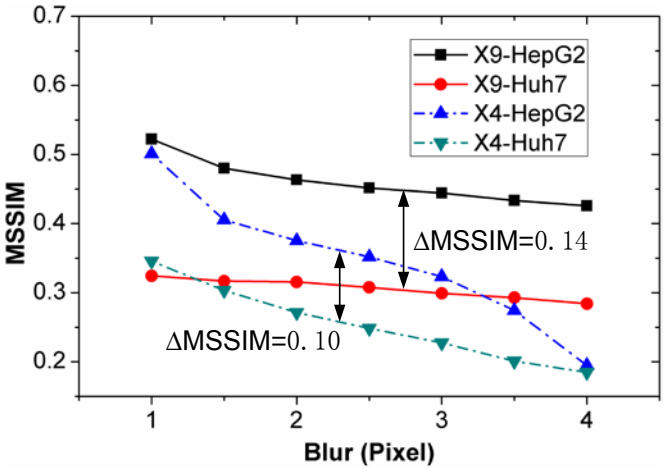

Fig. 12. The comparison of MSSIM at different magnification factor and blur effects.

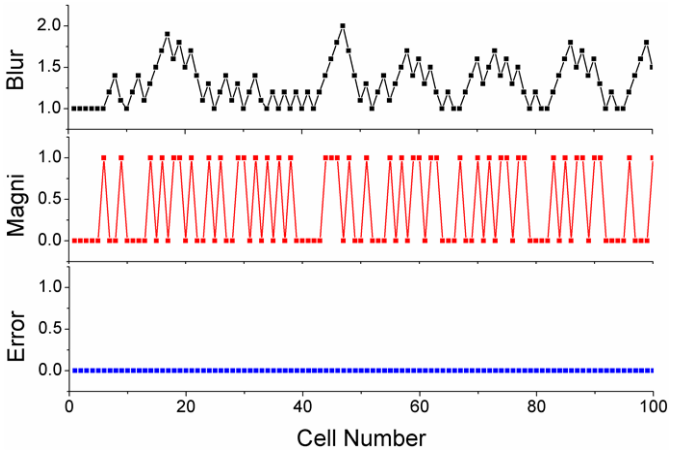

Fig. 13. The control results of blur, magnification factor, and recognition error for a series of cell inputs.

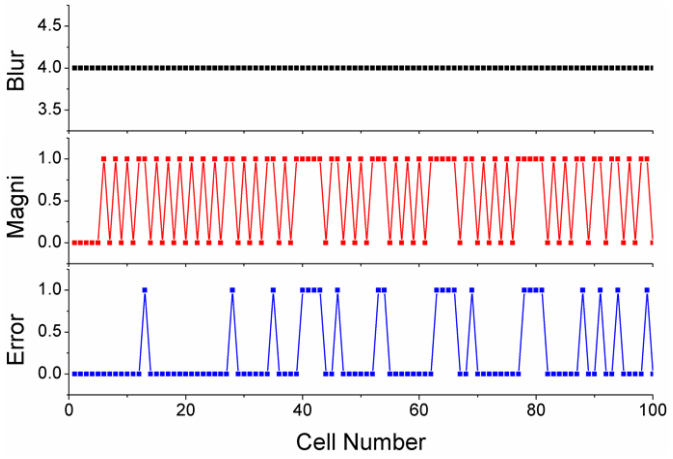

Fig. 14. The results of magnification factor tuning only without flow rate control, and recognition error for a series of cell inputs.

Table 1.
Notations and terminologies.

Notation	Parameter
$I_{m\times n}$	LR lensless image
$I_{M\times N}$	HR, HF, LR_int and SR lensless image
m, n	Row and column number of LR image
M, N	Row and column number of HR image
F_{mag}	Magnification factor
R	Pump flow rate
V	Cell flow speed
F_{ps}	Sensor frame rate
t_{exp}	Exposure time
H	Microfluidic channel height
W	Microfluidic channel width
D_{pix}	Sensor pixel size
R_{es}	Spatial resolution
p	Number of blurred pixel
t	Number of training images in one library
C_{tr}	Flow rate control cycle
$\boldsymbol{T}$	Training target value
$\boldsymbol{X}$	Training input data set

Table 2
Number of addition and multiplication in F_{mag}=4 mode.

	ADD	MULTIPLY	SIGMOID
A·X	13×20×900	14×20×900	0
AX+B	20×900	20×900	0
G(**AX+B**)	0	0	20×900
$\boldsymbol{\beta}$ **·H**	19×900	20x900	0
LR_Int+HF	32×32	0	0
Summary	270124	288000	18000

Table 3
Number of addition and multiplication in F_{mag}=9 mode.

	ADD	MULTIPLY	SIGMOID
A·X	13×20×4900	14×20×4900	0
AX+B	20×4900	20×4900	0
G(**AX+B**)	0	0	20×900
$\boldsymbol{\beta}$ **·H**	19×4900	20x4900	0
LR_Int+HF	72×72	0	0
Summary	1470284	1568000	98000

Table 4.
The comparison of resource utilization between FPGA implemented multi-frame SR and single-frame SR.

SR Method	Processing Blocks	Slice Registers	Slice LUTs	Block RAM
Multi-Frame SR	Control	110	263	0
	Bilinear Interpolation	1026	25510	0
	Motion Estimate	7099	78762	0
	LR Frames map	2540	4027	0
	Interpolation	3350	36187	0
	Median Filter	4109	9719	0
	RAM Storage	0	0	294
	Total	18234	154468	294
Single-Frame SR	Interpolation	469	672	0