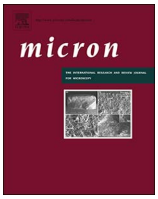
Micron 107 (2018) 55–71

Contents lists available at ScienceDirect

Micron

journal homepage: www.elsevier.com/locate/micron



Fast and robust segmentation of white blood cell images by self-supervised learning

Xin Zheng[a,*], Yong Wang[b], Guoyou Wang[c], Jianguo Liu[c]

[a] The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, School of Computer and Information, Anqing Normal University, Anqing 246133, China
[b] Department of Computer Science and Technology, The Hong Kong University of Science and Technology, Hong Kong, China
[c] National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Keywords:
Cell segmentation
Self-supervised learning
Support vector machine
White blood cell
Automatic labeling of training data

ABSTRACT

A fast and accurate white blood cell (WBC) segmentation remains a challenging task, as different WBCs vary significantly in color and shape due to cell type differences, staining technique variations and the adhesion between the WBC and red blood cells. In this paper, a self-supervised learning approach, consisting of unsupervised initial segmentation and supervised segmentation refinement, is presented. The first module extracts the overall foreground region from the cell image by K-means clustering, and then generates a coarse WBC region by touching-cell splitting based on concavity analysis. The second module further uses the coarse segmentation result of the first module as automatic labels to actively train a support vector machine (SVM) classifier. Then, the trained SVM classifier is further used to classify each pixel of the image and achieve a more accurate segmentation result. To improve its segmentation accuracy, median color features representing the topological structure and a new weak edge enhancement operator (WEEO) handling fuzzy boundary are introduced. To further reduce its time cost, an efficient cluster sampling strategy is also proposed. We tested the proposed approach with two blood cell image datasets obtained under various imaging and staining conditions. The experiment results show that our approach has a superior performance of accuracy and time cost on both datasets.

1. Introduction

White blood cells (WBCs or leukocytes), which are the principal components of immune cells, play a vital role in the fight against infections. In clinical practice, the identification and counting of WBCs in blood smear are often used for diagnosing many diseases such as infections, inflammation, malignancy, leukemia, etc. In the past, the examination of blood smears is a highly complex, tedious, and time-consuming manual task. Nowadays, with the rapid development of computer-aided methods, an automatic cell analysis system can support faster and more reproducible image analysis than manual analysis (Xing and Yang, 2016). Automatic cell analysis system generally includes four steps: image acquisition, cell segmentation, feature extraction and classification. Cell segmentation is often considered as the most important and critical step in the process, as it directly affects the accuracy and time complexity of subsequent steps.

A typical blood smear image consists of WBCs, red blood cells (RBCs or erythrocytes), platelets and the background. The goal of cell segmentation is to extract WBCs from such complicated scenes and provide essential information for the feature extraction step. By using the salient color of nuclei, WBC detection has been well solved by various cell segmentation methods (Ko et al., 2009; Ko et al., 2011; Zheng et al., 2014). An example of WBC detection result is shown in Fig. 1. However, fast and robust segmentation of WBCs remains challenging. The reasons are three-fold. First, the original blood smear images are significantly different in color due to different staining techniques and illumination conditions. Second, variations may exist even within the same smear image because of different types of WBCs. For example, WBCs can usually be classified into five types (i.e., lymphocytes, monocytes, neutrophils, eosinophils and basophils) and different types of WBCs stained by the same technique display various colors. As shown in Fig. 2, the colors of different sub-images vary significantly due to both inter- and intra-image variations of the original blood smear images. Thirdly, WBCs frequently adhere to RBCs, leading to irregular shapes of WBCs, and the boundaries between the touching cells are blurred. Therefore, an accurate WBC segmentation is a

---

[*] Corresponding author.
E-mail addresses: zxaoyou@hust.edu.cn (X. Zheng), ywangct@cse.ust.hk (Y. Wang), gywang@hust.edu.cn (G. Wang), jgliu@ieee.org (J. Liu).

https://doi.org/10.1016/j.micron.2018.01.010
Received 4 August 2017; Received in revised form 25 January 2018; Accepted 26 January 2018
Available online 01 February 2018
0968-4328/ © 2018 Elsevier Ltd. All rights reserved.

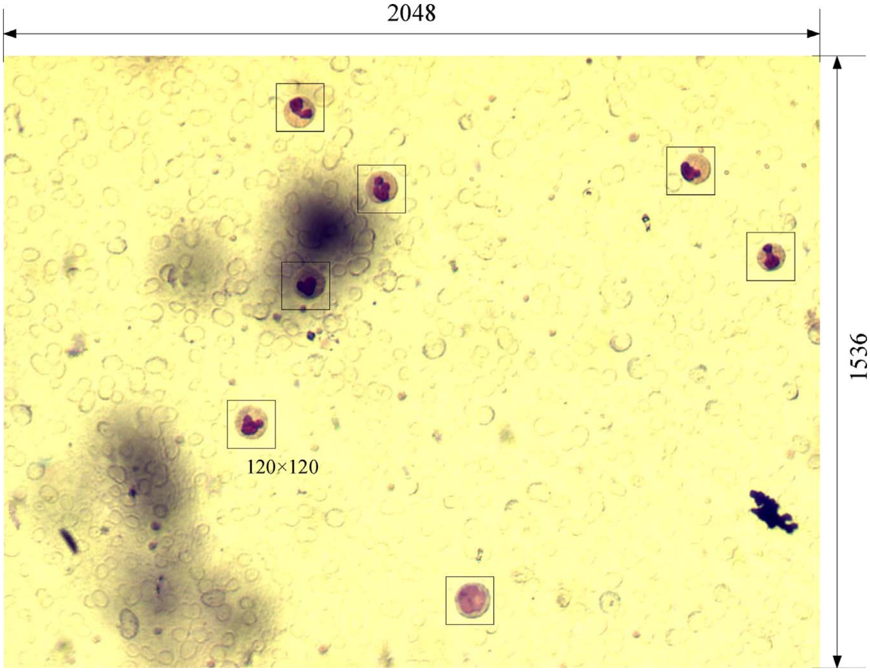


Fig. 1. WBC detection result by using an earlier work (Zheng et al., 2014).

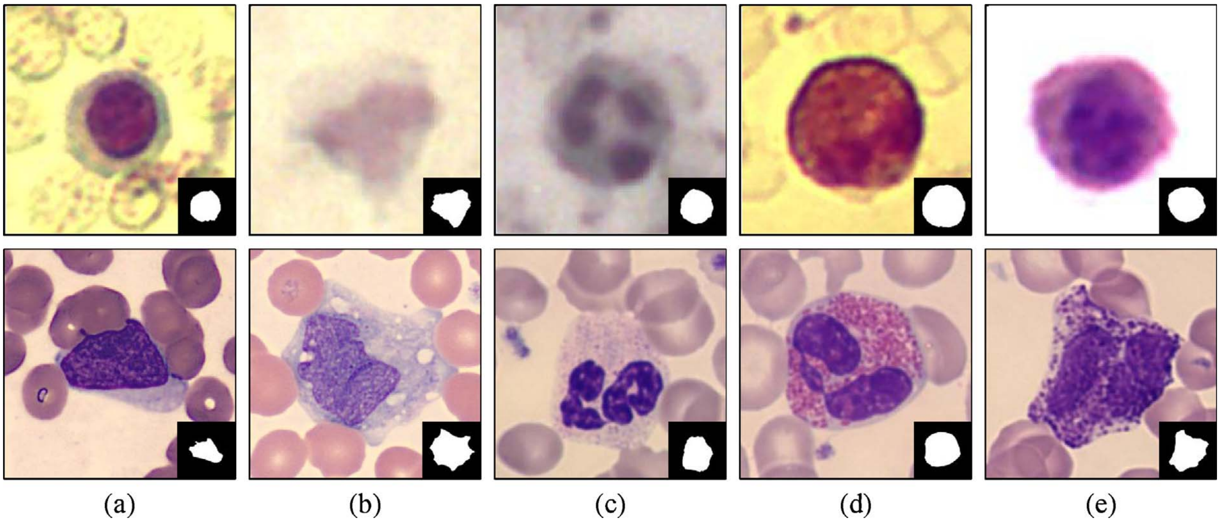


Fig. 2. Different types of WBC under various imaging and staining conditions. (a) Lymphocytes. (b) Monocytes. (c) Neutrophils. (d) Eosinophils. (e) Basophils. Top row: rapidly-stained images. Bottom row: traditional Wright-stained images. The ground truth segmentation results manually sketched by expert hematologists is shown at the bottom right corner of each image.

challenging task because of the above three reasons. In this paper, we propose a self-supervised learning approach to achieve fast and robust WBC segmentation and focus on segmenting the cell region of interest (CROI), i.e., the whole WBC region including both nucleus and cytoplasm, from RBCs and the background which is denoted as the non-cell region of interest (non-CROI).

Various automated cell segmentation methods have been developed. Most of them are learning-based methods, which can generally be classified as supervised and unsupervised methods. The supervised methods, formulate the problem of segmentation as a multi-class classification. For example, each pixel can be classified into CROI and non-CROI. Classifiers such as Bayesian (Prinyakupt and Pluempitiwiriyawej, 2015), K-nearest neighbor (Kong et al., 2011), neural network (Yi et al., 2005), support vector machine (SVM) (Song et al., 2013; Ruberto et al., 2016), and random forest (Saidi et al., 2016), etc., have been used, where the classifiers are usually trained on manually-labeled training images. Their effectiveness highly depends on the imaging conditions and whether the extracted features can distinguish the CROI from the non-CROI (Song et al., 2013). Thus, some researchers try to extract more effective color features to mitigate this issue, for example, use L*a*b* color space (Sertel et al., 2009) or the most discriminant color space (MDC) (Kong et al., 2011), adopt scale-invariant feature transform (SIFT) descriptor (Song et al., 2013), and integrate texture features (Farhan et al., 2013a). Apart from using the traditional classifiers, deep convolutional networks (Ronneberger et al., 2015) have also recently been applied in cell image segmentation. These methods based on deep convolutional networks usually can get better results than the traditional classifier-based learning methods. However, all the above-mentioned supervised methods need a large number of training samples, which are usually manually labeled but hard to gained for biomedical images. Besides, the supervised approaches usually do not work well when there are significant differences between the training and test samples, though they can produce good cell segmentation when the color distributions of both CROI and non-CROI are similar to those of training images.

The unsupervised learning methods, such as thresholding (Huang et al., 2012), K-means (Zhang et al., 2014), fuzzy c-means (FCM) (Theera-Umpon, 2005), and mean-shift (Zhi et al., 2015), etc., where manually labeled training data is not necessary, have stronger adaptability to various conditions than the supervised learning methods. They are effective when the color of the entire cell is distinct from others. However, they do not perform well when the CROI has significant color variations. Since a WBC has similar color with RBCs and often adheres to the RBCs, lots of parameters need to be adjusted case-by-case to avoid over- and under-segmentation. To overcome this problem, some clustering-based cell segmentation methods first separate the background from the foreground (including the CROI and RBCs) and then remove RBCs by shape-based methods, according to the fact that cells usually have a convex shape. There are mainly three types of shape-based methods: region-based Watershed (Xiaowei et al., 2006; Yang et al., 2006; Arslan et al., 2014), basic morphological operations (Dorini et al., 2007) and contour-based concavity analysis (Kumar et al., 2006; Yang et al., 2008; Farhan et al., 2013b). The concavity analysis methods, which are less sensitive to cell size than the former two methods, have drawn more and more attention. However, gray/color intensity information is not considered in the shape-based methods and they may probably fail when the cell shape is highly irregular.

In addition to the learning-based methods, there is another kind of powerful cell segmentation methods, called deformable models (Yang et al., 2005; Yan et al., 2008; Ko et al., 2011; Rezatofighi and Soltanian-Zadeh, 2011). These methods find the boundaries of the CROI by introducing contour evolution based on the computed internal and external forces. However, they suffer from the overlap and fuzzy boundaries between the WBC and RBCs and highly depends on the initial contour. If the initial contour is far from the real one, the final contour will probably converge to the boundaries of RBCs. Many deformable models try to accurately estimate the initial contours in a manner that is similar to that of traditional supervised (Yang et al., 2005) and unsupervised (Yan et al., 2008; Ko et al., 2011; Rezatofighi and Soltanian-Zadeh, 2011) learning methods. However, they will still have the same problems as other supervised and unsupervised algorithms do. Moreover, even if the initial contour is well positioned, the final contour may converge to the nucleus or RBCs when the boundary of the CROI is blurred or the contract between the CROI and non-CROI is low.

The major contribution of this paper is introducing a self-supervised learning approach to achieve fast and robust segmentation of white blood cells. The self-supervised learning approach consists of two modules (i.e., unsupervised initial segmentation and supervised segmentation refinement), making use of the advantages of both supervised and unsupervised methods. The rest paper is organized as follows. An overview of the proposed self-supervised learning approach is given in Section 2. The unsupervised initial segmentation and the supervised segmentation refinement, which are the two modules of the proposed method, are described in detail in Sections 3 and 4, respectively. Section 5 shows the experimental results. Then the discussion is presented in Section 6. The paper ends with our conclusions in Section 7.

2. Overview

The proposed cell segmentation approach is based on self-supervised learning, as shown in Fig. 3. It consists of two modules: unsupervised initial segmentation and supervised segmentation refinement.

In the first module, the background separation obtains the overall foreground region (a single CROI or a touching-cell clump) by using K-means clustering to remove the uniform background area. Then, we use a touching-cell splitting method to split the touching-cell clump roughly into the CROI and RBCs base on shape-based concavity analysis. The touching-cell splitting method is a two-step process: find all concave points on the contour of the touching-cell clump, then iteratively split the clump by connecting concave point-pairs with a straight line. Since no color or edge information is taken into account in splitting the touching-cell clump, the segmentation result is usually not good enough and false positive and negative segmentation results often exist, as shown in Fig. 3, which need to be refined by the second module.

The second module consists of three steps: feature extraction, sample selection and supervised classification.

In a supervised learning approach, each training sample contains an input feature vector and a label. Once the feature vector and label are known, a classifier can be trained actively. Therefore, the first step of the second module is feature extraction, whose goal is to feed pixel-wise feature vectors as the input to the classifier. Taking the local topological structure and fuzzy boundary problem into consideration, we introduce a 9-dimension descriptive feature vector, containing RGB color values, HSV median color values within a local neighborhood and HSV WEEO (weak edge enhancement operator) values.

If the features and labels of all pixels in the image were used as the training samples, it would take much time to train the classifier. To accelerate the classifier training process, an efficient cluster sampling strategy is proposed to sample only a small portion of representative pixels for the subsequent learning process.

The third step of the second module is supervised classification, which includes two phases: training and test. In the training phase, an SVM classifier is actively trained on the sample pixels that are automatically selected by the proposed cluster sampling strategy. In the test phase, the trained SVM classifier is used to classify each pixel of the image into the CROI or the non-CROI and achieve a more accurate segmentation result.

The proposed self-supervised learning approach combines supervised and unsupervised methods and outperforms them. Compared to the supervised learning methods, the proposed self-supervised learning approach is fully automatic with no human involvement and is more robust to color variations across images, as SVM classifier is trained on each image. Compared to the unsupervised learning methods, it can provide more accurate segmentation result while fewer parameters are needed to be tuned.

3. Unsupervised initial segmentation

The unsupervised initial segmentation consists of two steps: clustering-based background separation and touching-cell splitting using concavity analysis, where the first step removes the majority of the background and the second step cuts out the RBCs adhering to the CROI. The final output of this module is a coarse CROI segmentation result, which is used for training the classifier in the subsequent supervised segmentation refinement.

3.1. Background separation

The images we deal with here are sub-images cropped by an earlier work (Zheng et al., 2014). Each sub-image contains only a single CROI, the possible RBCs and other background. The CROI usually has color variations resulting from the imaging and staining conditions and the colors of the CROI and RBCs are often quite similar. Therefore, clustering methods using a fixed number of clusters are prone to produce over-and under-segmentation results. Different from the touching-cell clump (CROI/RBCs) with various colors, the background of the sub-images has higher brightness and relatively uniform color, making it easier to remove the background. Therefore, we adopt the classical K-means clustering to separate the background and retain the CROI region.

K-means clustering needs a pre-defined cluster number, K. Since

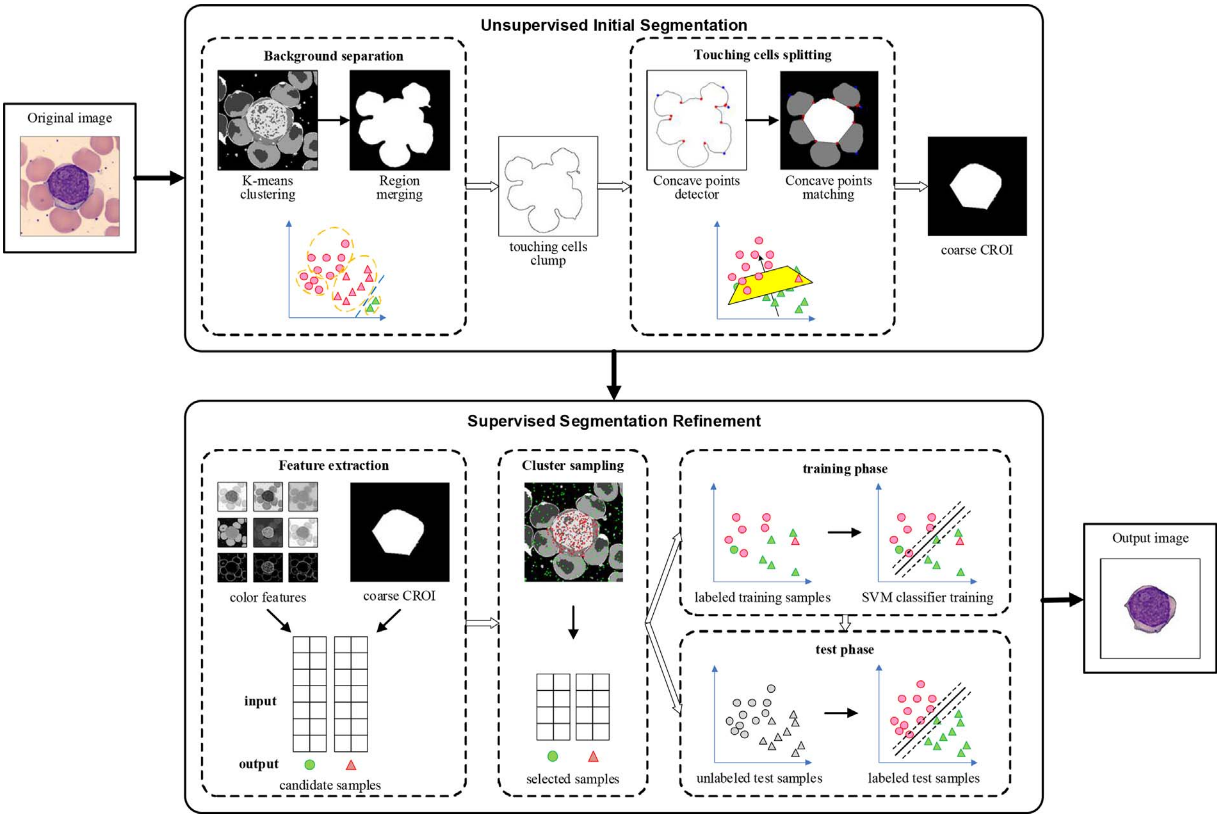


Fig. 3. Overview of our self-supervised learning approach.

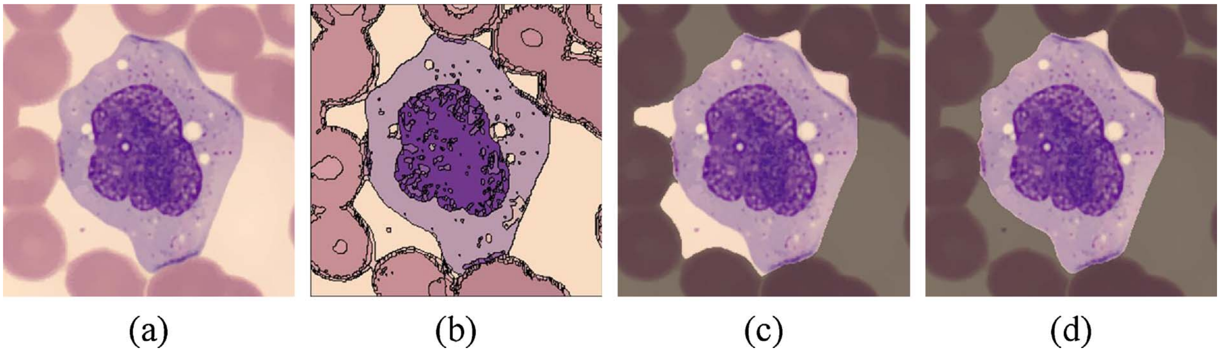


Fig. 4. Illustration of background separation. (a) A sub-image. (b) The over-segmentation result using K-means clustering. (c) The result of removing the regions at the border of the sub-image. (d) The result of further removing the regions that have similar color with the background. The removed regions in (c) and (d) are shaded.

there are at least four clusters in the cell image, which corresponds to the nucleus, cytoplasm, RBC and the background, K should be bigger than four to prevent the algorithm from segmenting blurred boundaries of the CROI into the background. The selection of the cluster number K will be discussed in Section 5.2. The background separation is done in the following four steps (One example is shown in Fig. 4).

(1) Convert the image (Fig. 4(a)) from RGB color space to HSI space and over-segment it into uniform regions R_i by K-means clustering (e.g., Fig. 4(b)).
(2) Remove the over-segmented regions located at the border of the image. Note that most of the background including some RBC areas have been removed after this step (see Fig. 4(c)). All the over-segmented regions that are removed after this step are labeled as B_i.
(3) Remove the regions that have very similar color with the background. Since B_i may contain both background and RBCs areas, the region $B_{\max}$ that has the maximum intensity value in B_i must be a background area. Therefore, the regions that have similar color with $B_{\max}$ can be regarded as the background and thus removed. The color similarity is defined as follows:

$$D(R_i, R_j) = ||C_i - C_j|| \tag{1}$$

where C_i is the average HSI vector of R_i and $\| \;\|$ indicates the Euclidean distance. According to the experiment, if $D(R_i, B_{\max}) < 10$, R_i can be empirically considered as a background area and should be removed, shown in Fig. 4(d).
(4) Merge the remaining regions (which are connected and usually located at the center of image) into the foreground, where the CROI are included.

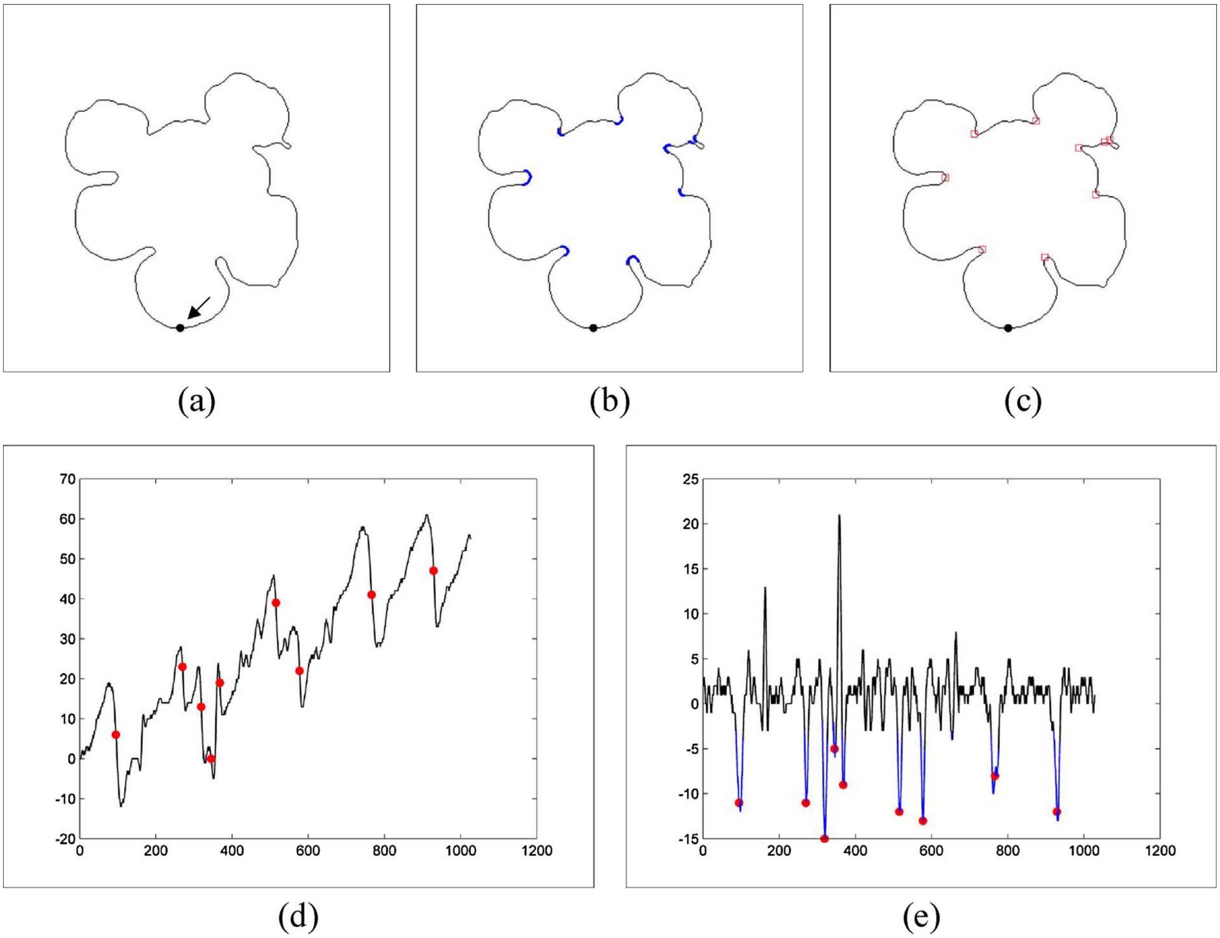


Fig. 5. Illustration of concave point detection. (a) The contour of a touching-cell clump. (b) The concave regions detected by the proposed method. (c) The concave points obtained by the proposed method. (d) The chain code sum of (a), which is calculated from the point marked by an arrow and in a counterclockwise direction. (e) The chain code difference of (a). The concave regions are highlighted in blue in (b) and (e) and the concave points are marked in red in (c), (d) and (e). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. *Touching-cell splitting*

The foreground extracted in the previous step may be a single CROI or a touching-cell clump that contains RBCs. However, due to the color similarity and spatial proximity between the CROI and RBCs, it is still very difficult to segment the touching-cell clump into the CROI and RBCs accurately. To automatically determine whether it is a single CROI or a clump and further decide how to split the clump, we propose a touching-cell splitting technique. Considering that cells are always convex, we make use of shape-based concavity analysis to check if there is a clump and further split the touching-cell clump. The touching-cell splitting technique consists of two steps: concave point detection and concave point matching.

3.2.1. *Concave point detection*

Let $C = \{p_i\}_{i=1}^{n}$ be the closed contour of the foreground, which is encoded using Freeman's chain code (Freeman, 1961), where p_i is the ith pixel on the contour, n is the number of the contour pixels and $i = 1, 2, \ldots, n$. The encoded chain can be described as $\{a_i\}_{i=1}^{n}$, where $a_i = \overline{p_i p_{i+1}}$, $a_i \in [0,7]$. To accurately estimate the slope of p_i, we use the chain code sum s_i of the adjacent $2k + 1$ points where the middle point is p_i (k is set to 3). Fig. 5(d) shows the curve of chain code sum of a touching-cell clump (Fig. 5(a)). To obtain the concave points, compute the chain code difference d_i, which means the difference of the chain code sum between s_{i+k} and s_{i-k}. The curve of chain code difference is shown in Fig. 5(e). According to the experiment, if $d_i \le -4$, the point p_i can be regarded as a concave point. Then, all the concave regions on the contour can be detected in this way (see Fig. 5(b)). The most likely concave points are found as the midpoints of corresponding concave regions (see Fig. 5(c)).

3.2.2. *Concave point matching*

Building off the concave point detection, we propose an iterative concave points matching technique to split a touching-cell clump if it exists, and further obtain a coarse CROI. At each iterative step, the two concave points with the local maximum arc-chord ratio (ACR) are matched as a pair, and the touching-cell clump is split into two by connecting the concave point-pair with a straight line. The ACR can be calculated as

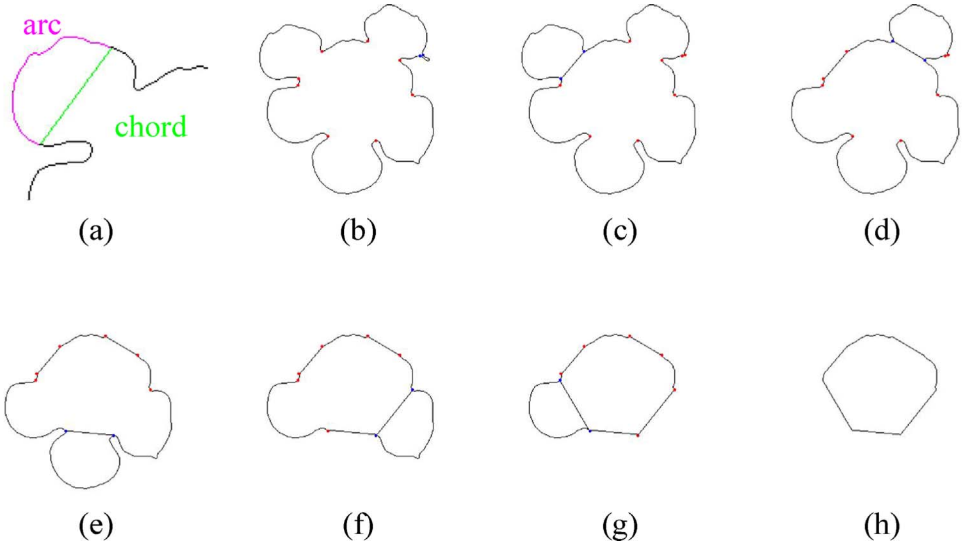
$$\gamma = \frac{L_e}{L} \tag{2}$$

where L_e is the effective length of a curve (marked in magenta in Fig. 6(a)), and L is the length of the straight line between the curve's two endpoints (marked in green in Fig. 6(a)). As shown in Fig. 6, the RBCs that adhere to the CROI are separated from the clump step by step. The details of concave points matching algorithm can be found in Algorithm 1.

Algorithm 1 Concave point matching

```
Input: A touching-cell clump and a set of concave points $\{P_i\}_{i=1}^{m}$
Output: The rough region of the CROI
$flag \leftarrow \mathrm{true}$;
while $flag$ do
    Calculate the ACR $\gamma$ between each pair of concave points $\{P_i, P_j\}, i \neq j$;
    Find the pair of concave points with the maximum $\gamma$, $\{P_{c_1}, P_{c_2}\} = \max_{\gamma}\{P_i, P_j\}$;
    if $\max(\gamma) \geq 2$ then
        Split the clump into two by cutting along the line connecting $P_{c_1}$ and $P_{c_2}$;
        Keep the central region while removing other surrounding regions;
    else
        $flag \leftarrow \mathrm{false}$;
    end if
end while
```

Fig. 7 gives some example results of the touching-cell splitting. However, since only the shape feature is considered and no color or edge information is taken into account, over- and under-segmentation may still exist after touching-cell splitting. This will be further improved by the following supervised segmentation module.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. Illustration of splitting a touching-cell clump using the iterative concave points matching. (a) Illustration of arc and chord. (b-h) The procedures of separating the RBCs that adhere to the CROI from the cell clump, where one cell is separated from the clump at each step.

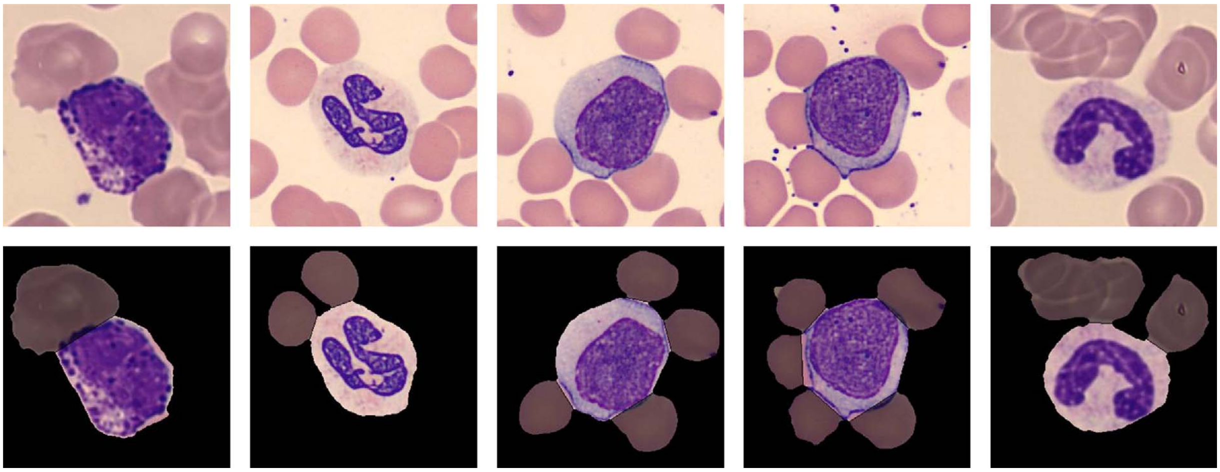


Fig. 7. Examples of touching-cell clump splitting. Top row: the original images. Bottom row: results of splitting the touching-cell clump, where the background removed in *background separation* is shown in black and the RBCs removed in *touching-cell clump splitting* are shaded.

4. Supervised segmentation refinement

Our self-supervised learning approach will train an SVM classifier and further classify each pixel into either the CROI or the non-CROI. As mentioned in Section 2, a supervised learning approach needs training samples where each sample contains an input feature vector and a label. The training samples are usually manually labeled. In contrast, our self-supervised learning approach makes the labeling of training samples totally automated by using the result of unsupervised initial segmentation.

Through background separation and touching-cell splitting, a coarse CROI is extracted from the input image and will be used as the labels to train the classifier. In the following subsections, we will discuss how to extract good features and select representative training samples to guarantee both the accuracy and efficiency of the segmentation.

4.1. Feature extraction

Extracting distinguishable features from various images is key to achieve optimal performance of the supervised learning approach. We consider both the color features and the topological structure information (Kontschieder et al., 2011) of each pixel, and further include a type of new features to improve the segmentation performance for handling the blurred cell boundaries.

Color feature is one of the most informative features in describing the cell image and RGB color values, $\boldsymbol{C}^{RGB} = (R, G, B)$, are often used to describe the image pixel. However, according to our experimental result, holes and non-continuous regions may always exist in the CROI if only the color information of each pixel is used. This is probably because each pixel is considered independently and the local topological structure (Kontschieder et al., 2011) is lost.

To incorporate the topological structure of the local neighborhood of each pixel, we calculate the median color value of all the surrounding pixels in a $W \times W$ neighborhood window (Fig. 8(e)), $\boldsymbol{MC}^{HSI} = (MC^H, MC^S, MC^I)$, which is calculated in HSI color space. By combining the neighboring median color feature and the pixel color feature, a more continuous CROI region can be obtained.

In addition, the blurred boundaries of WBCs, where the intensity contrast between the CROI and the non-CROI may be rather low, usually lead to inaccurate segmentation of WBCs (either under-segmentation or over-segmentation). To handle this issue, we introduce a new type of features for each pixel based on a weak edge enhancement operator (WEEO), which can enhance the weak boundaries between the CROI and the non-CROI and support better segmentation. Inspired by a prior work (Arbeláez et al., 2011), the proposed WEEO is also based on a circular template with alterable directions. The template is centered at O and divided into two half-discs by a diameter of D pixels at angle θ, as shown in Fig. 8(i). The detailed steps of extracting WEEO-based features for each pixel are described as follows:

(1) Compute the oriented gradient between the histograms (say, g and h) of the two half-discs at angle θ, which is defined as below:

$$G_\theta(g, h) = \sum_{i=1}^{l} \left\{ d(g_i, h_i) \Big/ \left[\sum_{j=1}^{l} d(g_j, h_j) - d(g_i, h_i) \right] \right\} \tag{3}$$

where i and j are the bin indices and l indicates the total number of bins in the histogram. $d(g_i, h_i)$ is calculated in the following way:

$$d(g_i, h_i) = \frac{(g_i - h_i)^2}{g_i + h_i} \tag{4}$$

where both g and h are normalized histograms. The oriented gradient mainly encodes the differences between the two half-discs from the perspective of histograms.

(2) Calculate the oriented variance of the intensity for each half-disc, as defined below:

$$V_{\theta,m} = \frac{1}{S_m} \sum_{i \in \Omega_m} (I_i - \overline{I_{\Omega_m}})^2 \tag{5}$$

where $m \in \{u, l\}$ denotes the upper or lower half-disc, Ω_m is the set of pixels falling into the same half-disc m, S_m represents the total number of pixels in Ω_m, I_i and $\overline{I_{\Omega_m}}$ indicate the intensity of pixel i and the mean intensity of the pixels in Ω_m, respectively. The closer the intensities of the pixels in Ω_m are to the mean intensity of the half-disc, the smaller the value $V_{\theta,m}$ is, and vice versa.

(3) Calculate the degree of the uniform distribution of intensities in both half-discs, as defined below:

$$V_\theta = 1 \Big/ \exp\left(\prod_m (V_{\theta,m} + 1) \right) \tag{6}$$

The more consistent the intensity of each half-disc, the larger the V_θ value is, and vice versa. The value of V_θ varies between 0 and 1.

(4) Calculate the product of the variance weight V_θ and the oriented gradient G_θ over eight directions in $[0,\pi]$. The WEEO value is the maximum product across all the directions, as defined below:

$$WEEO = \max_\theta \{G_\theta \times V_\theta\} \tag{7}$$

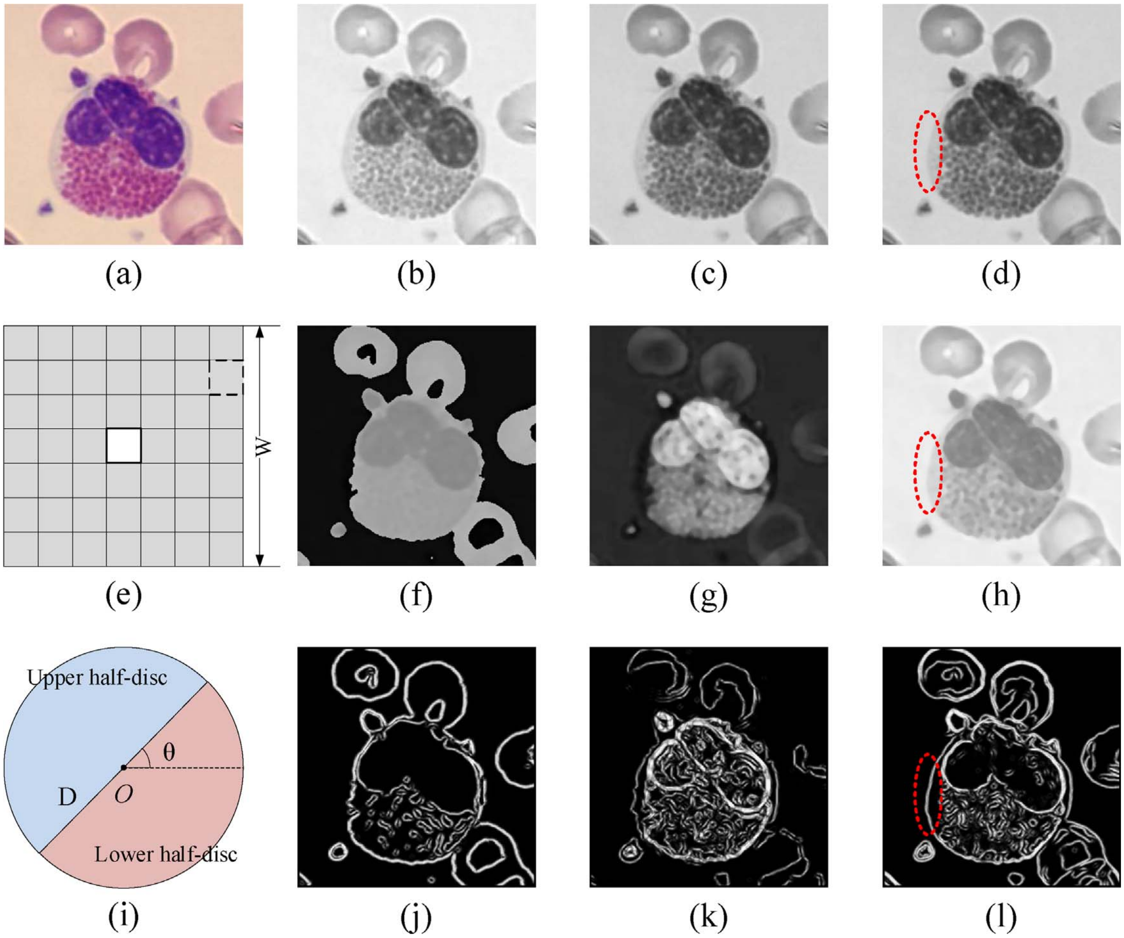


Fig. 8. Visual representation of color features. (a) The original image, (b)-(d) are the R, G, B component of color features respectively. (e) A $W \times W$ ($W = 7$) neighboring window. (f-h) are the H, S, I component of median color feature ($W = 7$) respectively. (i) The circular template of WEEO. (j-l) are the H, S, I component of features calculated by WEEO ($D = 7$) respectively. The weak boundary is marked in red dotted ellipse. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Compared with the contour detector introduced in (Arbeláez et al., 2011), the intensity variance is used as the weight (as shown in Eq. (7)) in the proposed WEEO. It is able to enhance the gradient of weak boundaries between CROI and non-CROI and the WEEO-based features support better segmentation for weak boundaries, as shown in the area marked in red dotted ellipse in Fig. 8. In this paper, we calculate the WEEO-based features in HSI color space for each pixel, i.e., $\boldsymbol{WEEO}^{HSI} = (WEEO^{H}, WEEO^{S}, WEEO^{I})$.

By now, a 9-dimensional feature vector $\boldsymbol{x} = [\boldsymbol{C}^{RGB}, \boldsymbol{MC}^{HSI}, \boldsymbol{WEEO}^{HSI}]$, which takes into account the color features, topological structures and enhancement of blurred boundaries, is obtained. Fig. 8 shows an example of these features. There are three parameters to be determined in the feature extraction procedure, namely, the size of the neighboring window W, the diameter D, and the bin number l. For convenience, we simply set $D = W$. The choice of these parameters will be discussed in Section 5.2.

4.2. Sample selection

It will take a lot of time to train a classifier if all the image pixels are used for training. To accelerate the classifier training process, we introduce an efficient cluster sampling strategy, where only a small number of representative training samples are chosen to train the classifier. These pixel samples are selected from the over-segmented regions obtained by K-means clustering in *background separation* and the over-segmented regions are already labeled as two types of regions (i.e., the CROI and the non-CROI) by the unsupervised initial segmentation. The sample size of each over-segmented region is proportional to its area size. More specifically, for each over-segmented region $R_i (i = 1, 2, ..., H$, H is the total number of over-segmented regions), the representative pixels are sampled as follows:

(1) Scan each row of pixels in R_i and create two arrays, $L_i^{+}[N_i^{+}]$ and $L_i^{-}[N_i^{-}]$, to store the locations of pixels in the CROI and the non-CROI respectively, where N_i^{+} and N_i^{-} are the total number of the CROI and the non-CROI pixels in R_i.

(2) Calculate the sample sizes S_i^{+} and S_i^{-} (i.e., the number of sampled CROI and non-CROI pixels in R_i) using the following formula:

$$S_i^{b} = 0.5S \times N_i^{b}/N^{b}, \quad b \in \{+,-\} \tag{8}$$

where N^{+} and N^{-} are the total numbers of the CROI and the non-CROI pixels in the whole sub-image, respectively. S is the pre-defined total number of sample pixels and we sample the same number of pixels (i.e. $0.5S$) from the CROI and the non-CROI.

(3) Compute the sampling intervals for the CROI and the non-CROI in all the over-segmented regions in this way:

$$I^{b} = N^{b}/0.5S, \quad b \in \{+,-\} \tag{9}$$

(4) Sample S_i^{+} pixels in $L_i^{+}[N_i^{+}]$ with an interval of I^{+} pixels and S_i^{-} pixels in $L_i^{-}[N_i^{-}]$ with an interval of I^{-} pixels. Note we will sample at least one pixel in each over-segmented region.

The selection of the total sampling size S and the corresponding

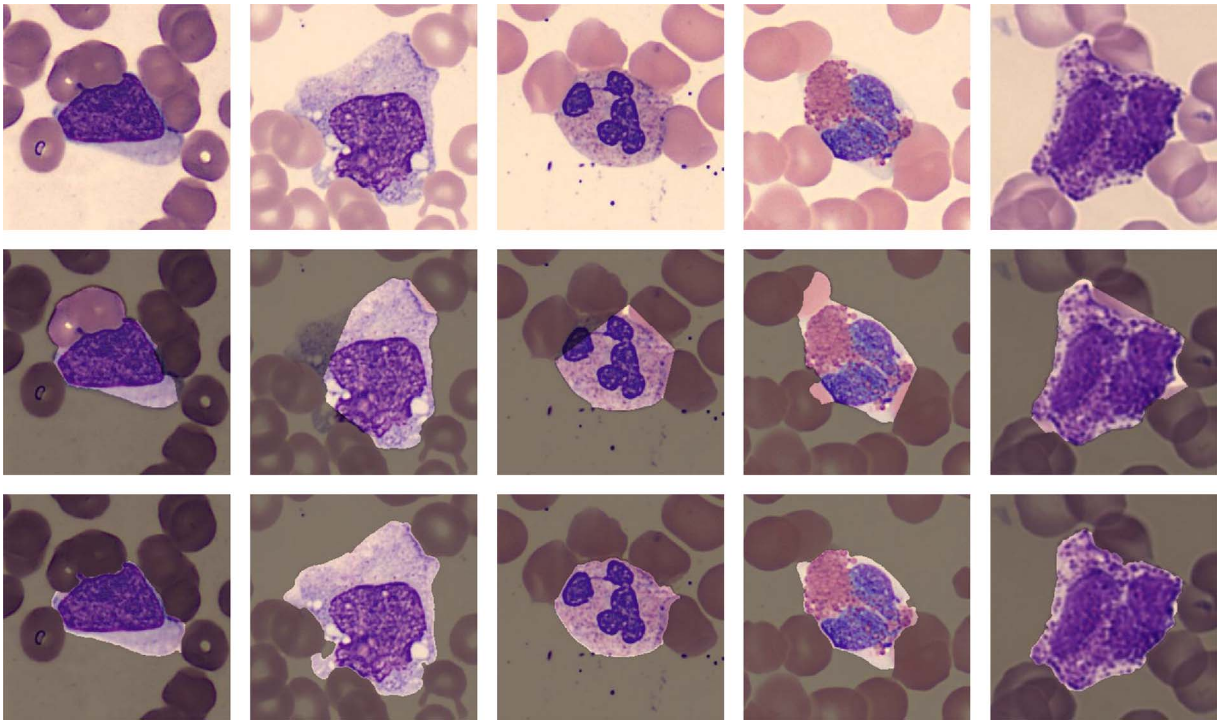


Fig. 9. Comparison of segmentation results with and without using the proposed supervised classification. Top row: the original images. Middle row: results after unsupervised initial segmentation. Bottom row: results after supervised segmentation refinement. The removed regions in the sub-images of middle row and bottom are shaded.

impact of adjusting S will be discussed in Section 5.2.

4.3. Supervised classification using SVM

Support vector machines (SVMs) have been widely used in many applications scenarios (Cristianini and John, 2000), especially in solving the classification and pattern recognition problem with a small number of training samples. Considering its excellent capability in classification, we also choose SVM as the classifier in our self-supervised learning approach. The SVMs try to find the hyperplane that maximizes the margin between two classes and its core idea can be briefly summarized as follows (Vapnik, 1998):

Suppose the training set is $\{\boldsymbol{x}_i, y_i\}_{i=1}^{S}$, where $\boldsymbol{x}_i$ is the input feature vector, $y_i = \pm 1$ is the binary labels (indicating the CROI and the non-CROI), S is the size of training samples and $i = 1, 2, \ldots S$. The decision function of SVM is

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{S} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right) \tag{10}$$

where α_i^* and b^* are the Lagrange multiplier and bias respectively that will be learned from the training dataset. $K(\boldsymbol{x}_i, \boldsymbol{x})$ is a kernel function, which defines the similarity between the input feature vector $\boldsymbol{x}$ and the support vector $\boldsymbol{x}_i$.

To find the optimal parameters α_i, we need to maximize the Lagrangian

$$\mathrm{W}(\alpha) = \sum_{i=1}^{S} \alpha_i - \frac{1}{2}\sum_{i=1}^{S}\sum_{j=1}^{S} \alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{11}$$

under the following constraints

$$\sum_{i=1}^{S} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C \tag{12}$$

where C is a penalty parameter to be specified by the user. A larger C corresponds to assigning a higher penalty to misclassification.

To apply SVM in our self-supervised learning approach, we also need to define the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x})$. The general kernel function choices for SVM include the linear kernel, the polynomial kernel, the radial basis kernel and the sigmoid kernel. In this paper, the radial basis function (RBF) kernel is chosen in the proposed approach and defined as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}\|^2) \tag{13}$$

where γ is a regularization parameter that can be tuned for optimal performance of the classifier. The selections of parameters C and γ will be discussed in Section 5.2.

Fig. 9 shows some examples with and without using the supervised classification: the middle row shows the result without using the supervised classification (i.e., the result of unsupervised initial segmentation), while the bottom row shows the segmentation results of supervised segmentation refinement. It can be clearly seen that over- and under-segmentation problems are solved by the supervised classification.

It should be noted that the training pixel samples fed into the classifier may contain some wrong labeled samples. However, the proposed self-supervised learning approach can still achieve a good segmentation accuracy. The main reason is that our well-designed unsupervised initial segmentation can guarantee that the wrong labeled samples are significantly less than the correctly labeled samples and most of the training samples have correct labels. The impact of these few wrong labeled pixels can be mitigated by the dominant samples with correct labels. Also, the proposed clustering-based sampling strategy supports a better selection of representative sample pixels and the novel pixel features can enable effective discrimination between the CROI and the non-CROI.

5. Results

We did extensive experiments on two image datasets to evaluate the proposed self-supervised learning approach. We tested its robustness to parameter variations, assessed the effectiveness of the proposed features and sampling strategy, and further compared our approach with the existing representative cell segmentation methods.

5.1. Materials and evaluation methods

All the experiments of this paper are based on two image datasets captured under different imaging and staining conditions by two medical microscopy companies. Dataset 1 was obtained from Jiangxi Tecom Science Corporation, China. It contains three hundred 120 × 120 sub-images of single WBC (176 neutrophils, 22 eosinophils, 1 basophil, 48 monocytes, and 53 lymphocytes), which are cropped from 80 source images by an earlier work (Zheng et al., 2014). The source images of Dataset 1 were taken by a Motic Moticam Pro 252A optical microscope camera with a N800-D motorized auto-focus microscope, and the smears were processed with a newly-developed hematology reagent for rapid WBC staining (Zheng et al., 2014). The size of each image is 2048 × 1536 and the color depth is 24 bits. Dataset 2 consists of one hundred 300 × 300 color images (30 neutrophils, 12 eosinophils, 3 basophils, 18 monocytes and 37 lymphocytes), which were collected from the CellaVision blog.[1] To evaluate the accuracy of the proposed approach and other cell segmentation methods, each image has a ground truth segmentation result indicating the CROI (white) and the non-CROI (black), which were manually sketched by expert hematologists. To benefit the research community of medical image segmentation, we have made both datasets publically available.[2]

Following prior work on cell segmentation, we used four popular segmentation error measures: one contour-based measure called *relative distance error* (*RDE*) (Yang-Mao et al., 2008) and three region-based measures including *over-segmentation rate* (*OR*), *under-segmentation rate* (*UR*), *overall error rate* (*ER*) (Yi et al., 2005; Chan et al., 2010; Pan et al., 2012; Zhang et al., 2014). *OR*, *UR* and *ER* are described as follows:

$$OR = Q_p/(U_p + D_p) \tag{14}$$

$$UR = U_p/(U_p + D_p) \tag{15}$$

$$ER = (Q_p + U_p)/D_p \tag{16}$$

where Q_p is the number of pixels that should be included in the segmentation result but are not, U_p is the number of pixels that should be excluded from the segmentation result but actually included, and D_p is the number of pixels that should be included in the segmentation result and are also actually included.

RDE is popularly used in the evaluation of object segmentation. Let $e_1, e_2, ...e_{n_e}$ and $t_1, t_2, ...t_{n_t}$ be the pixels on E and T respectively (where E and T are respectively the extracted contour pixels and the target contour pixels) and n_e and n_t the number of pixels on E and T. *RDE* is thus defined as:

$$RDE = \frac{1}{2}\left(\sqrt{\frac{1}{n_e}\sum_{i=1}^{n_e} d_{e_i}^2} + \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t} d_{t_i}^2}\right) \tag{17}$$

where $d_{e_i} = \min\{dist(e_i, t_j)|j = 1, 2, ..., n_t\}$, $d_{t_j} = \min\{dist(e_i, t_j)|i = 1, 2, ..., n_e\}$, and $dist(e_i, t_j) = ||e_i, t_j||$ represents the Euclidean distance between e_i and t_j.

The smaller the values of the four segmentation errors, the better the segmentation performance. The time costs of each segmentation methods are also evaluated.

5.2. Evaluation of robustness to parameter variations

In this subsection, we show that the proposed approach is robust to parameter variations by presenting the effects of different parameters on the final segmentation results. There are several parameters that need to be determined in the proposed self-supervised learning approach: (1) the number of clusters K in *Background Separation*, (2) the total sampling size S in *Sampling Selection*; (3) the size, W, of the neighboring window $W \times W$ in *Feature Extraction*; (4) the bin number l in *Feature Extraction*; and (5) the kernel type and related parameters of SVM in the *Supervised Classification*.

Fig. 10(a–d) show the effects of the first four key parameters in this study. It can be seen that, when a single parameter changes its value within a certain range while the other model parameters are set as the empirically-chosen constant values, which will be introduced below, the segmentation performance remains nearly constant. Similarly, as shown in Fig. 10(e–f), the four segmentation error measures do not change significantly when the parameters γ and C of SVM vary significantly. It is clear that the proposed approach shows strong robustness to parameter variations. Taking both segmentation accuracy and calculation speed into account, we have empirically chosen the following parameter values for all the other experiments in this paper: (1) $K = 8$; (2) $S = \max(w, h)$ where w and h are width and height of the image, respectively; (3) $W = 7$; (4) $l = 8$; and (5) $\gamma = 1$, $C = 100$.

5.3. Evaluation of feature effectiveness

To verify the effectiveness of the proposed color features, we compared the final segmentation results using three sets of color features: the RGB color values $\boldsymbol{C}^{RGB}$, the feature set $[\boldsymbol{C}^{RGB}, \boldsymbol{MC}^{HSI}]$, and the proposed color feature set $[\boldsymbol{C}^{RGB}, \boldsymbol{MC}^{HSI}, \boldsymbol{WEEO}^{HSI}]$, which are further denoted as “C”, “C + MC” and “C + MC + WEEO”, respectively.

As shown in Fig. 11(f–g), “C + MC” produces a more accurate segmentation result than that of “C” on both datasets, which can be seen from the decreased *OR*, *ER* and *RDE*. For example, there is a decrease of 10.8%, 4.6%, 1.5% in *OR*, *ER* and *RDE* respectively on Dataset 1 and a decrease of 20.1%, 9.2%, 4.3% on Dataset 2. The result shows that the added median color features in HSI color space can increase the robustness of the proposed approach to the holes and non-continuous regions, which can be seen from the comparison between Fig. 11(c) and (d). Moreover, “C + MC + WEEO” further reduces the segmentation errors compared with “C”. For instance, there is 6.8% decrease in *ER* on Dataset 1 and 14.5% decrease on Dataset 2. It confirms the effectiveness of the proposed features in handling blurred boundaries and an example for this is shown in the second row of Fig. 11(a–e), where the blurred boundary is marked in a red dotted ellipse.

5.4. Evaluation of sampling effectiveness

In this subsection, we assess the effectiveness of the proposed clustering sampling (CS), through comparing it with the generally-used uniform sampling (US). Due to the limited space and the consistency between the four segmentation error metrics, we only evaluate their segmentation errors using *RDE*. As shown in Fig. 12(g), when the total sampling size is the same, the segmentation errors in *RDE* using CS are significantly reduced compared to that using US, with only a slightly increased time cost (11%–26%). It demonstrates that CS can effectively sample more representative training sample pixels and thus achieve better segmentation results than US. Also, it can be seen from Fig. 12 that the final segmentation result using CS (Fig. 12(f)) is comparable to the ground truth (Fig. 12(d)), although the training samples may contain noise (see Fig. 12(c)).

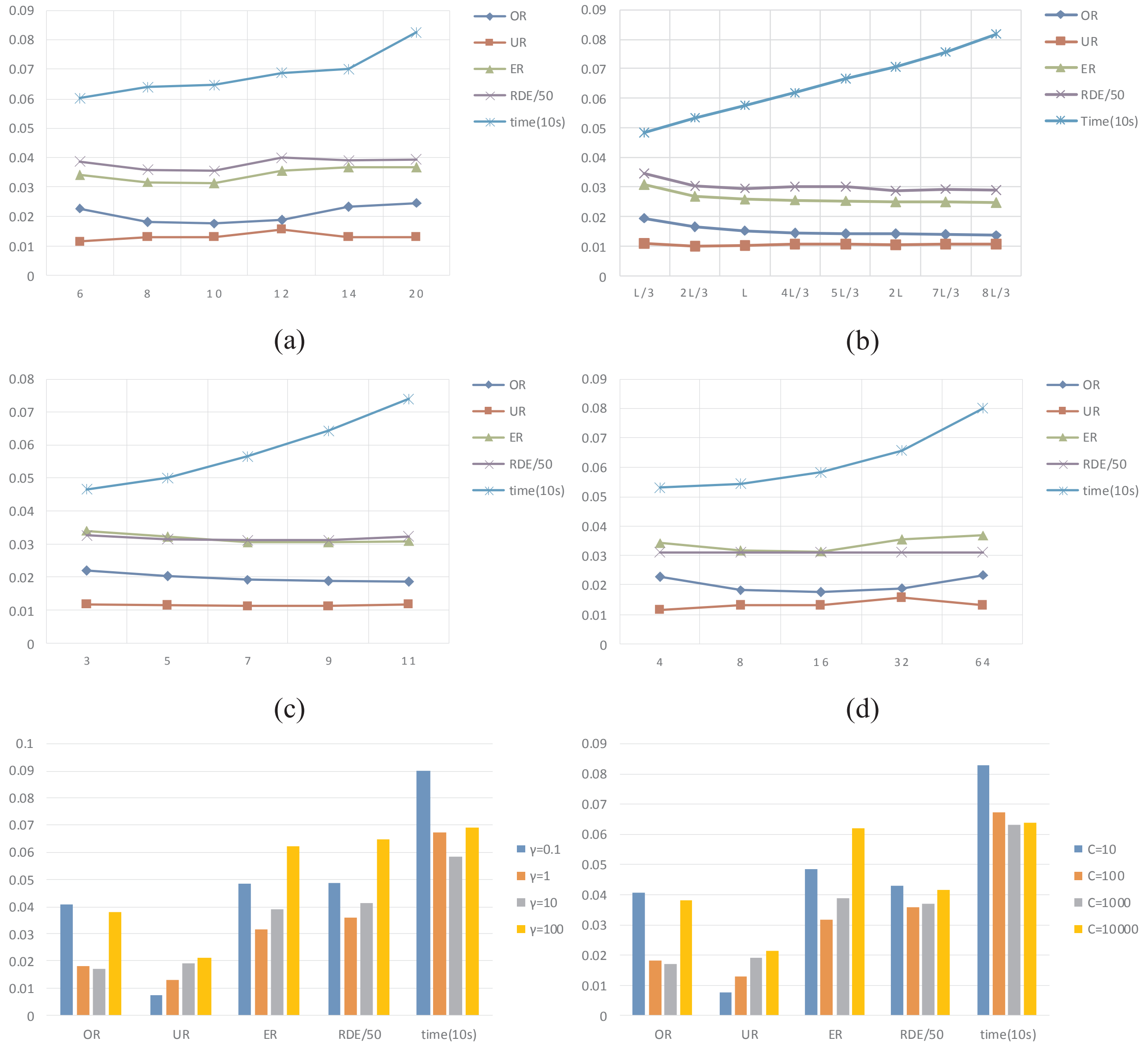
5.5. Comparison of segmentation performance

To further evaluate the overall segmentation performance of our self-supervised learning approach, we have compared it with four existing cell segmentation methods. Three of them are unsupervised methods, namely, fuzzy c-means (FCM) (Theera-Umpon, 2005), canny-based gradient vector flow snake (CGS) (Ko et al., 2011), simulated visual attention (SVA) (Pan et al., 2012). The fourth method is a convolutional neural network approach named U-Net (Ronneberger et al.,

---

[1] http://blog.cellavision.com.

[2] https://github.com/zxaoyou/segmentation_WBC.



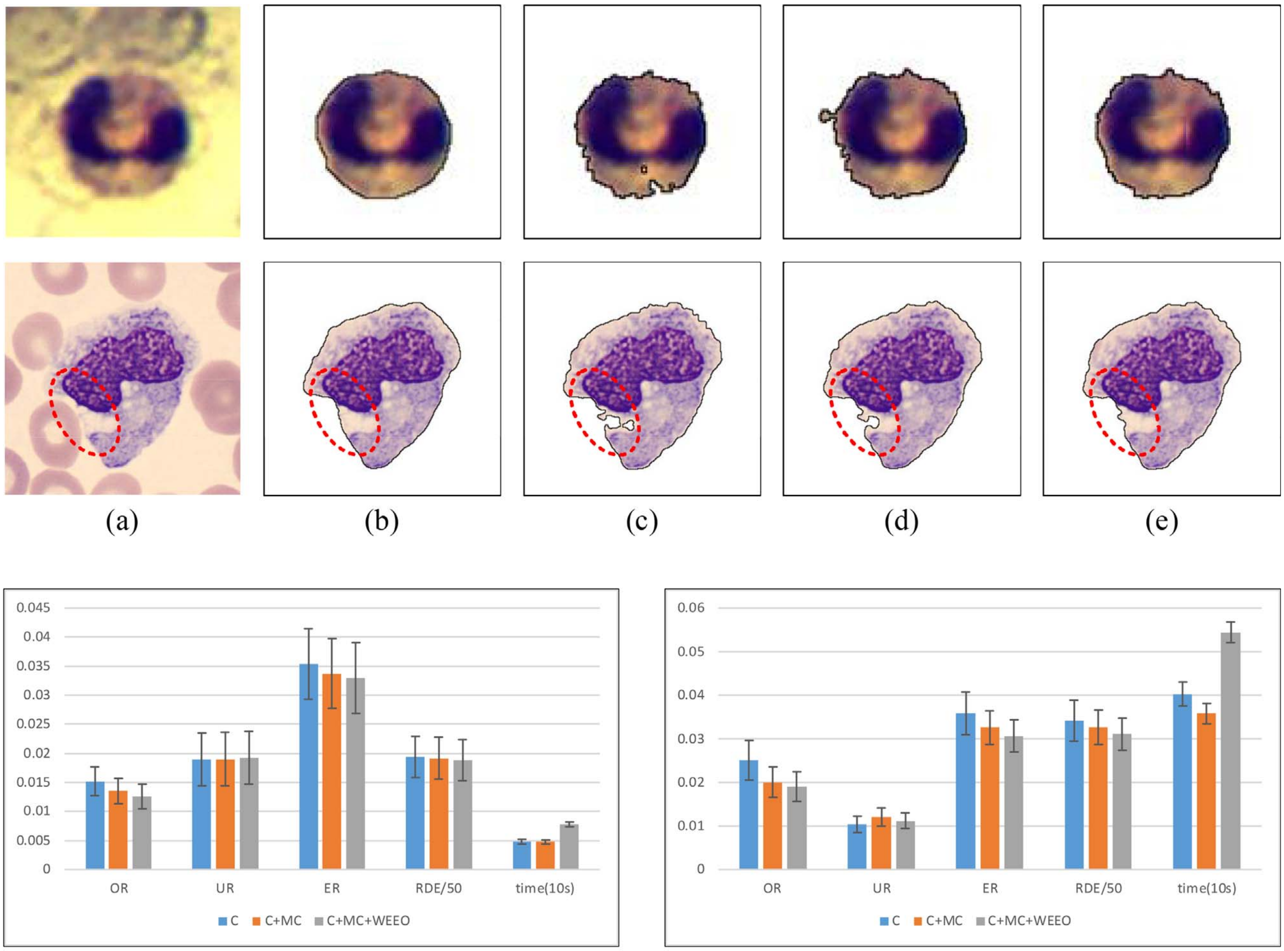
(a) (b) (c) (d) (e) (f)

Fig. 10. The effect of the parameters used in the proposed approach. (a) The number of clusters K in *background separation*. (b) The total sampling size S in *sample selection*. $L = \max(w, h)$, where w and h are the width and height of the input sub-image. (c) The size W and (d) the bin number l in *feature extraction*. (e) Parameter γ of RBF kernel in SVM. (f) Parameter C of SVM. To display all the four segmentation error measures and the time cost intuitively in a single figure, we replace RDE with RDE/50, and the time units are ten seconds, the same for the subsequent figures.

2015), which is a supervised learning method. We run 10-fold cross-validation experiments to evaluate the performance of U-Net. We equally divided all the images (which are resized to 300 × 300) of both datasets into 10 folders and further used images of one folder as the test data and the images of the remaining folders as the training data. Tables 1 and 2 show the average segmentation errors. We tried various parameters to train a U-Net model with the best segmentation performance. It was optimized by using the momentum-based stochastic gradient descent. The U-Net model with the least training loss was gained with a learning rate of 0.06, a weight decay of 0.995 and a momentum of 0.2. The total iterations for the training are set as 40,000 (1000 epochs for 40 iterations/epoch). All the experiments were done on a desktop with 3.46 GHz Intel i5-4590 processors and 8GB RAM. To make a fair comparison of the time cost between different algorithms, the U-Net model is tested by using only CPU.

We compared the four existing segmentation methods with our approach on both datasets. Figs. 13 and 14 provide some segmentation examples on the two datasets. Tables 1 and 2 give the segmentation errors and time costs on Datasets 1 and 2, respectively. In addition, to explore their segmentation accuracy for different types of cells, we also present the segmentation performance corresponding to each type of WBCs, as shown in Fig. 15.

In general, our segmentation approach outperforms the others in terms of segmentation accuracy. On Dataset 1, our approach gives better results in most of the measures, as shown in Table 1 (the best results are highlighted in bold). Though our method is higher in UR than SVA (4.27% vs. 3.34%), our segmentation approach achieves better overall segmentation results in all the other measures. Particularly, our approach has the lowest *OR*, *ER* and *RDE* (0.69%, 5.24%, 1.16) compared with FCM (0.85%, 12.41%, 2.44), CGS (4.87%, 8.59%, 1.85) SVA (3.32%, 7.09%, 1.68), and U-Net (0.83%, 5.39%, 1.25). As for the segmentation results on Dataset 2, Table 2 demonstrates that our



(a) (b) (c) (d) (e)

(f) (g)

Fig. 11. Comparison of segmentation results using different sets of features. (a) The original sub-images. (b) The ground truth. (c) Results using "C". (d) Results using "C + MC". (e) Results using "C + MC + WEEO". Top row: an image from Dataset 1 and segmentation results using different sets of color features. Bottom row: an image from Dataset 2 and segmentation results. The blurred boundary is marked in red dotted ellipse. (f) Comparison results using Dataset 1. (g) Comparison results using Dataset 2. The error bar of (f) and (g) indicates the 95% confidence interval. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

approach has the lowest segmentation errors in all the four measures, exhibiting a much better segmentation performance with 49.7%, 55.3%, 58.4% and 46.7% decrease in *OR*, *UR*, *ER* and *RDE* respectively over the second best method (i.e., U-Net) in each metric. When comparing the running time, our method performs best among all the four approaches on both datasets.

The FCM method functions well on Dataset 1, especially with OR measure, which is only 0.85 and close to our method. However, FCM performs poorly in the rest of the measures. When the sub-images contain WBCs with uniform intensity and no other cells (e.g., RBCs) appear (Fig. 13(c)), FCM has a good segmentation performance. But when many RBCs exist in the sub-images or the color is unevenly distributed, as shown in Fig. 14(c), FCM is not able to remove RBCs and leads to over-segmentation for WBCs with blurred boundaries.

By replacing the gradient map with canny edge map, the contour-based CGS method improves the accuracy of contour detection for the cells with fuzzy boundaries. However, the generated contour by CGS may converge to the inner textures (e.g. the nuclei in the third row of Fig. 13(d)) or the outer textures (e.g., the non-CROI in the fourth row of Fig. 13(d) and the RBCs shown in the fifth row of Fig. 14(d)), when the boundaries of WBCs are blurry and the inner and outer textures have higher gradient. Besides, CGS is the most time-consuming method among all the four methods, as shown in Tables 1 and 2. For example, the average time cost of CGS on Dataset 2 is more than 50 s (56800 ms).

The classification-based SVA introduced by (Pan et al., 2012) is an online-classification-based method by sampling high gradient pixels around the nuclei. It is robust to color variation, but it may lead to over-segmentation, for example, the existence of holes inside cells (e.g. Fig. 14(e)), because of its improper sample labels.

Though the SVM classifiers in SVA and our approach are both trained automatically, ours outperforms SVA, which can be seen in both the segmentation examples (Figs. 13 and 14) and the quantitative accuracy comparisons (Tables 1 and 2). One major reason for this is that our approach can automatically select training samples that are more representative. The carefully-designed unsupervised module can obtain an acceptable initial segmentation result and our cluster sampling well preserves the important details of the CROI and non-CROI in the training samples. The other major reason is that the proposed color features can better describe the topological structure and deal with weak boundaries than the simple RGB color features used in SVA. Therefore, our approach can produce more continuous CROI and handle the weak boundaries of the CROI much better than SVA.

U-Net has achieved precise segmentations on many biomedical images, but needs only a few images for training. Though it is not originally designed for WBC segmentation, U-Net still yields good segmentation results on our datasets. However, our approach slightly outperforms U-Net in OR, ER and RDE, as shown in Tables 1 and 2. One possible reason for this is that the color of different images appears quite different and the color of CROI may be similar to that of a non-CROI across images or even within the same image, due to the variation

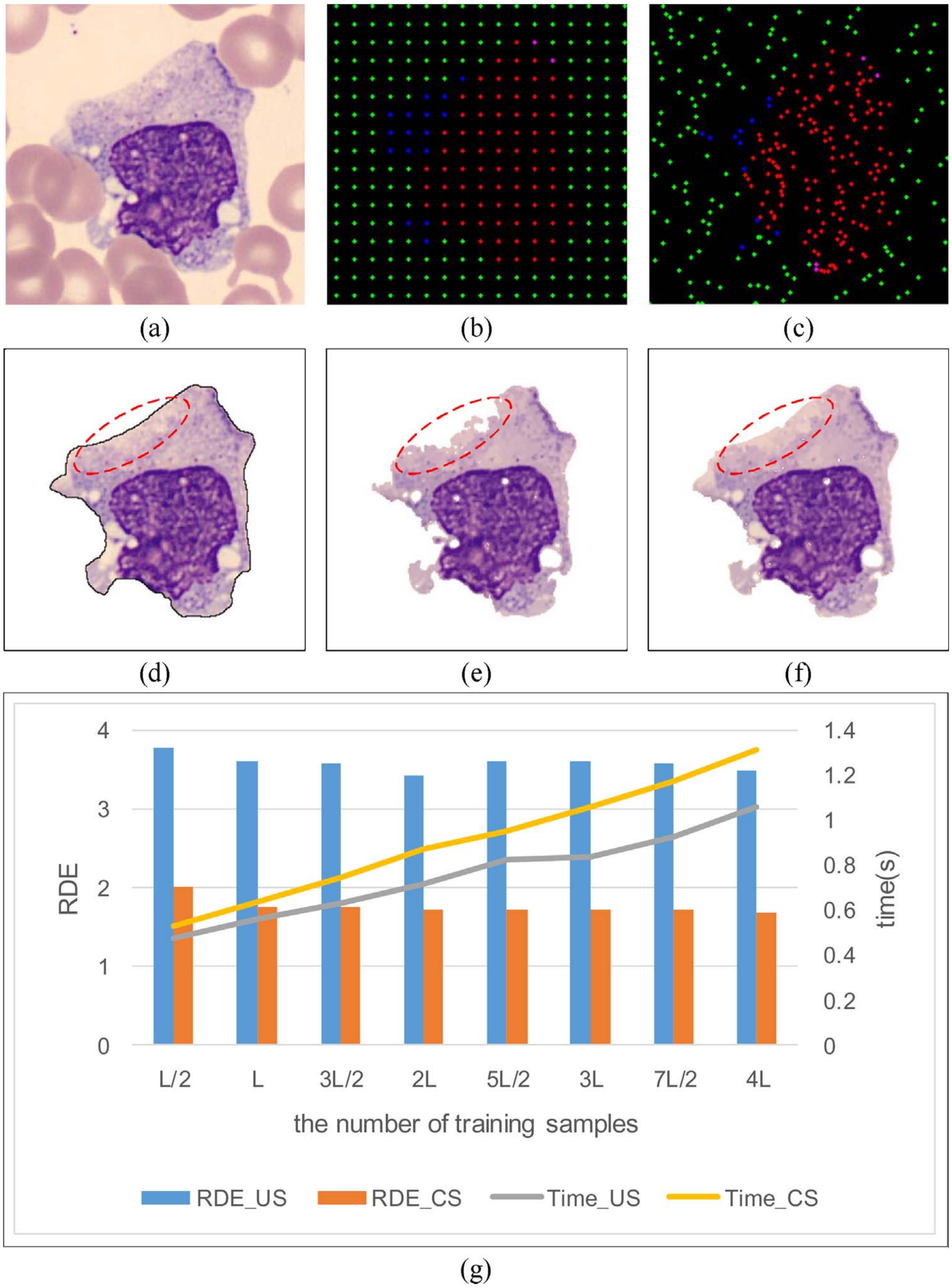


Fig. 12. Comparison of segmentation results using different sampling strategies. (a) The original image. (b) Uniform sampling (US). (c) Clustering sampling (CS). (d) The Ground truth. (e) The result of US. (f) The result of CS. Red points denote true positive training samples, green points represent true negative ones, magenta points are wrong positive ones and blue points are wrong negative ones. (g) Comparison of segmentation results using different sampling strategies on Dataset 2. $L = \max(w, h)$, where w and h are the width and height of the input sub-image. The blurred boundary region of (d-f) is marked in red dotted ellipse. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of the segmentation results on Dataset 1. (The smaller the value of OR, UR, ER and RDE, the better the segmentation performance. The best results are highlighted in bold).

	OR(%)	UR(%)	ER(%)	RDE	Time(ms)
FCM	0.85	9.94	12.41	2.44	273
CGS	4.87	3.37	8.59	1.85	880
SVA	3.32	**3.34**	7.09	1.68	139
U-Net	0.83	4.17	5.39	1.25	233[a]
Ours	**0.69**	4.27	**5.24**	**1.16**	**90**

[a] The time cost of U-Net is tested when using only CPU for a fair comparison, the same below.

Table 2
Comparison of the segmentation results on Dataset 2.

	OR(%)	UR(%)	ER(%)	RDE	Time(ms)
FCM	12.27	21.33	55.42	15.07	2260
CGS	5.17	7.31	14.05	6.74	56800
SVA	13.76	3.71	18.87	7.04	774
U-Net	3.72	2.93	7.64	3.36	1055
Ours	**1.87**	**1.31**	**3.18**	**1.79**	**639**

The best results are highlighted in bold.

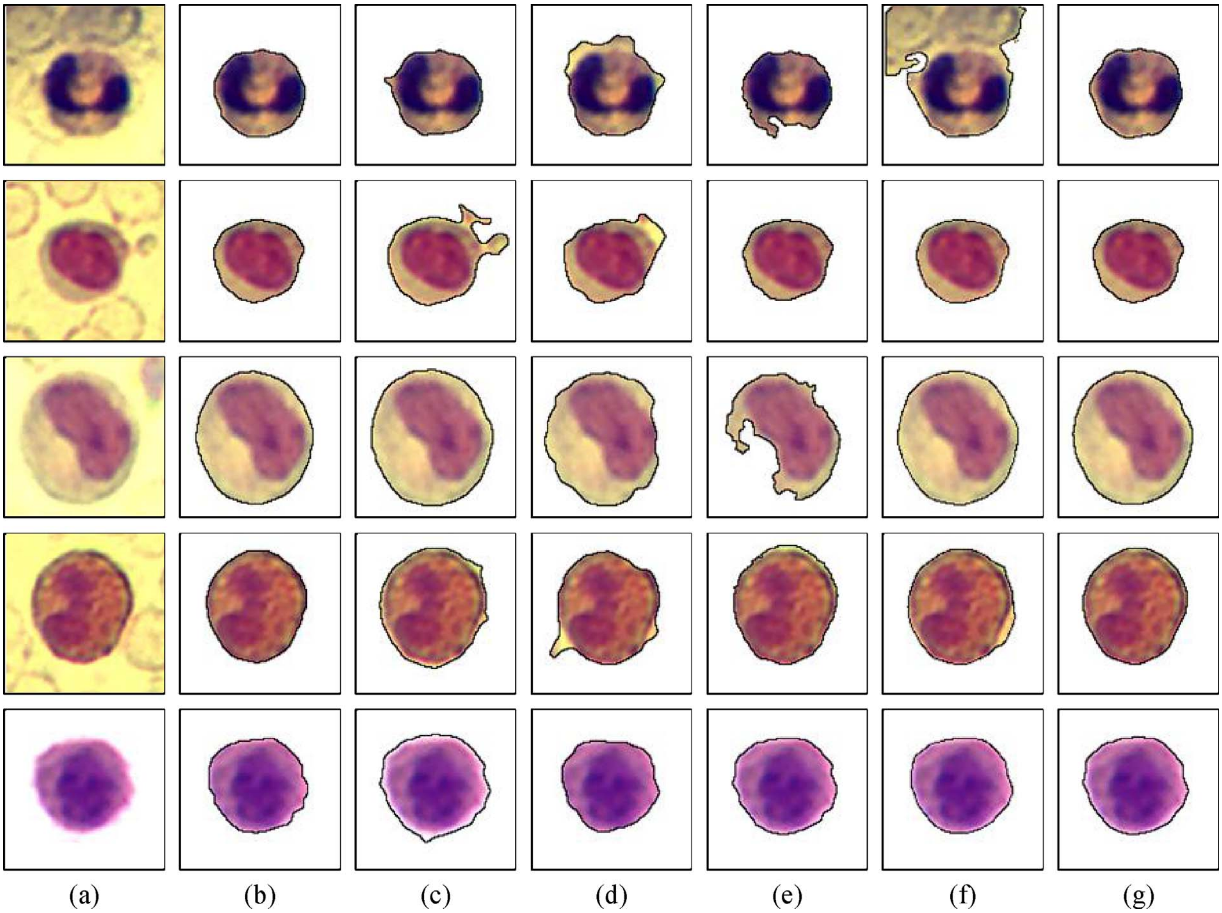


Fig. 13. Example segmentation results on Dataset 1. From top to bottom: neutrophils, lymphocytes, monocytes, eosinophils and basophils. (a) The original sub-images containing different types of WBCs. (b) The ground truth contours. (c) Results of FCM. (d) Results of CGS. (e) Results of SVA. (f) Results of U-Net. (g) Results of our self-supervised learning approach.

of staining techniques and illumination conditions. For example, the color of some parts of the background is quite like that of the neutrophil cytoplasm, leading to under-segmentation (the first row of Fig. 13(f)). Similarly, there is no significant color difference between the eosinophil cytoplasm and the background, leading to over-segmentation (the fourth row of Fig. 14(f)). U-Net may not be able to learn these subtle differences effectively. In addition, the performance of deep learning neural network highly depends on parameter tuning and finding the optimal parameters is a time-consuming task. In contrast, the proposed self-supervised learning approach can more precisely segment the WBCs by training and testing on the same input image. It is also easy to tune the parameters and more robust to parameter variations. For the time cost, when U-Net is running on only CPU for a fair comparison, it shows no advantages in time cost over the proposed approach. But it should be noted that the time cost of U-Net will be significantly reduced when using GPU.

6. Discussion

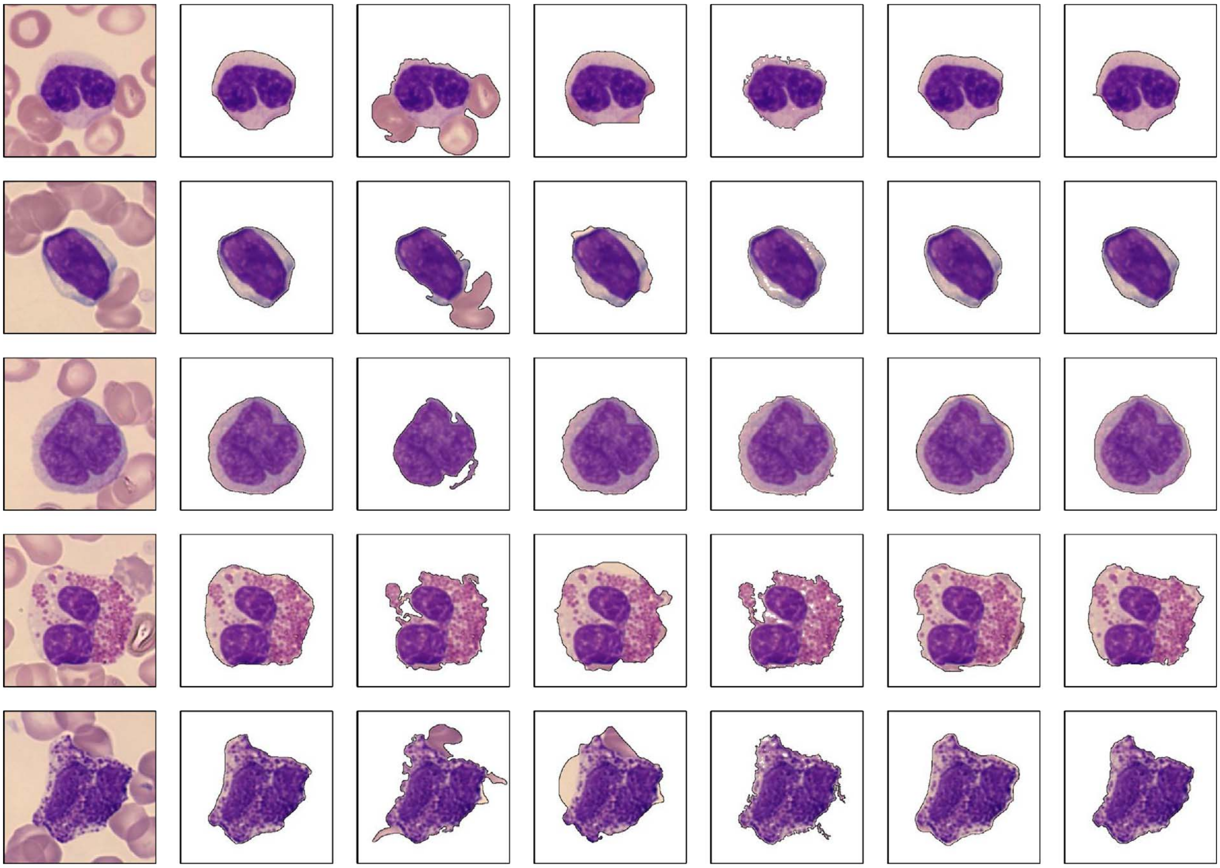
This paper introduces a self-supervised learning approach to the segmentation of white blood cells, where both unsupervised and supervised learning methods are involved. Its excellent segmentation performance and efficiency are also demonstrated through the extensive experiments above. However, there are still three major aspects that need further discussion.

6.1. *Sub-images vs. original large image*

In this paper, we only deal with sub-image that includes one single WBC and is cropped from the original image with a large size. This is different from some prior work (Pan et al., 2012; Zhi et al., 2015), that is designed for directly doing cell segmentation on the original large image. Two factors are considered here. First, there are many mature WBC detection algorithms, which can guarantee the detection accuracy and be used for obtaining sub-images from the original large cell images. An earlier work (Zheng et al., 2014) is chosen in this paper to crop sub-images from peripheral blood cell images, but it can also be replaced by other cell detection algorithms. Second, the overall processing speed of cell image segmentation is highly improved by working on the sub-images. As shown in Fig. 1, WBCs are scattered in the original large image and the majority of the image is the background. When focusing on only the region of interest, much time is saved. Therefore, this paper is designed for the cell segmentation of sub-images and it can be easily extended to cell segmentation of the original large images by combining one cell detection algorithm with the proposed self-supervised learning approach.

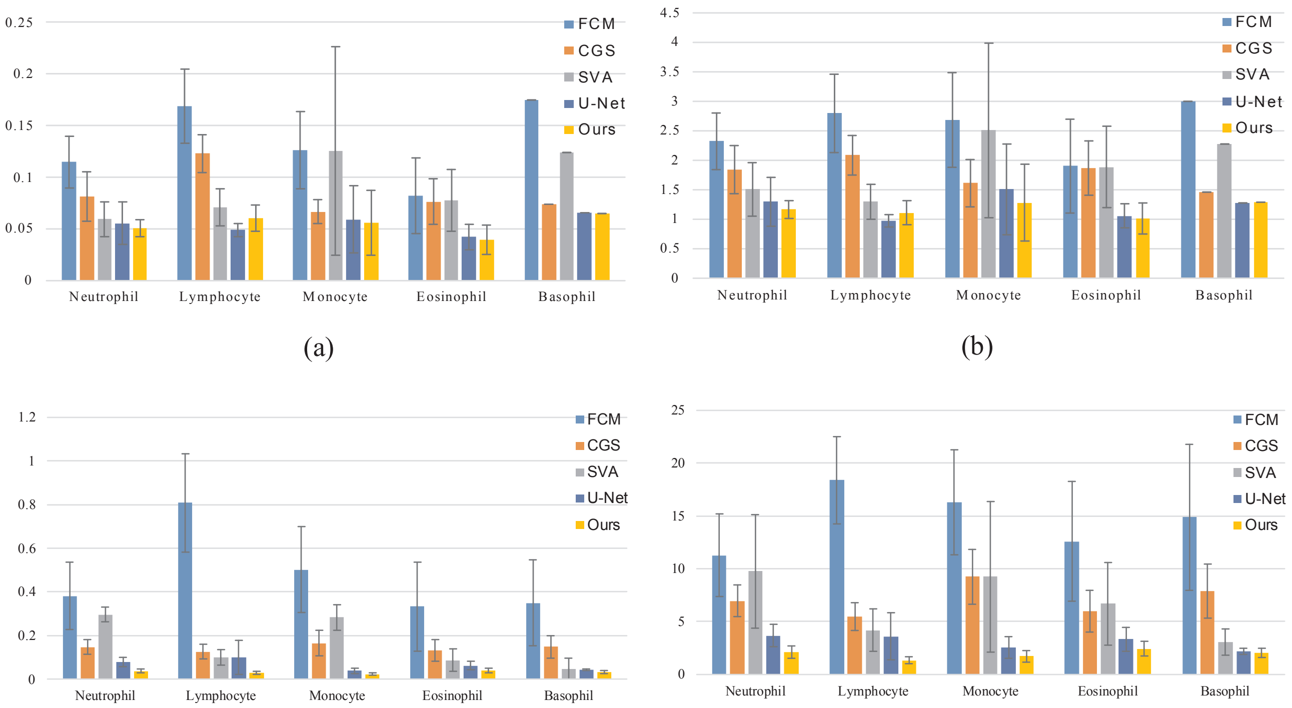
6.2. *Training every time vs. training once for all*

Due to the cell type differences, unstable staining and varying illumination, different sub-images vary in the intensity and color even within the same original cell image. Therefore, it is necessary to train one individual SVM classifier for each sub-image. Our experiment has shown that the time cost of all the training and test is still acceptable and better than the existing methods. However, the proposed approach can be easily adapted to further improve its efficiency by obtaining training samples from all the sub-images and training only once instead of training every time for every sub-image. This is especially useful for



(a) (b) (c) (d) (e) (f) (g)

Fig. 14. Example segmentation results on Dataset 2. From top to bottom: neutrophils, lymphocytes, monocytes, eosinophils and basophils. (a) The original sub-images containing different types of WBCs. (b) The ground truth contours. (c) Results of FCM. (d) Results of CGS. (e) Results of SVA. (f) Results of U-Net. (g) Results of our self-supervised learning approach.



(a) (b) (c) (d)

Fig. 15. Segmentation error rates of the four WBC segmentation methods on the five types of WBCs. (a) ER by using Dataset 1. (b) RDE by using Dataset 1. (c) ER by using Dataset 2. (d) RDE by using Dataset 2. The error bar indicates the 95% confidence interval.

the situation when the sub-images have consistent intensity and color. When the intensity and color are inconsistent across the sub-images, there is a trade-off between segmentation accuracy and better efficiency to train every time for each sub-image or train once for all.

6.3. Extension to other application scenarios

The experiment of this paper is mainly focusing on the segmentation of WBC in peripheral blood smear images. However, our self-supervised learning approach can also be extended to the segmentation of cells in other types of biomedical images. For example, by using one cell detection algorithm to generate the sub-images, our approach can potentially be applied in the cell segmentation of cervical smear images (Yang-Mao et al., 2008) and bone marrow smear images. More testing on these types of datasets is left as future work.

7. Conclusions

We present a self-supervised learning approach to improve the accuracy and adaptability of white blood cell segmentation. It consists of two major modules: the unsupervised initial segmentation generates a rough segmentation result and the supervised segmentation refinement uses these initial segmentation results to train an SVM classifier and achieve much-improved segmentation result. In addition, three important strategies, which further improve the segmentation accuracy and efficiency, are also proposed and integrated in our approach: (1) a fast touching-cell splitting algorithm, (2) an effective cluster sampling strategy, and (3) a novel color feature vector that includes median color feature to incorporate topological structure and the feature using a novel weak edge enhancement operator (WEEO) to improve the segmentation of fuzzy boundaries. The touching-cell splitting algorithm improves the adaptability of our approach, making it work for images under various imaging and staining conditions. The cluster sampling strategy can effectively select a portion of representative training samples, which guarantees both the segmentation accuracy and efficiency. The proposed color features represent more inherent properties of each sample pixel, thus are robust to non-continuous regions and blurred boundaries in the images.

Our experiments have demonstrated the effectiveness and advantages of the proposed self-supervised learning approach in white blood cell segmentation. It is robust to inter- and intra-image variations and is not very sensitive to parameter variation. The proposed features and sampling strategy can further improve the segmentation accuracy using the SVM-based classification. Compared with existing representative methods, our approach also shows much better segmentation performance.

In future work, we plan to further test the performance of the proposed approach on more datasets in various conditions. Also, we would like to apply the proposed self-supervised learning approach in an automated cell analysis system to achieve better identification and counting of WBCs.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (61603003), the Natural Science key Research Project for higher education Institutions of Anhui Province (KJ2016A439, KJ2016A438), Anhui Provincial Natural Science Foundation of China (1608085MF144) and the Foundation of University Research and Innovation Platform Team for Intelligent Perception and Computing of Anhui Province. The authors would like to thank Beifang Yi for his great help in the revision of the manuscript, and Chen Pan for his constructive suggestions on this project. The authors also thank Jiangxi Tecom Science Corporation in China and CellaVision Company in Sweden for sharing their image datasets.

References

Arbeláez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 33, 898–916.

Arslan, S., Ozyurek, E., Gunduz-Demir, C., 2014. A color and shape based algorithm for segmentation of white blood cells in peripheral blood and bone marrow images. Cytometry 85, 480–490.

Chan, Y.K., Tsai, M.H., Huang, D.C., Zheng, Z.H., Hung, K.D., 2010. Leukocyte nucleus segmentation and nucleus lobe counting. BMC Bioinf. 11, 558.

Cristianini, N., John, S.-T., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.

Dorini, L.B., Minetto, R., Leite, N.J., 2007. White blood cell segmentation using morphological operators and scale-space analysis. Computer Graphics and Image Processing, 2007. XX Brazilian Symposium on. IEEE 294–304.

Farhan, M., Ruusuvuori, P., Emmenlauer, M., Rämö, P., Dehio, C., Yli-Harja, O., 2013a. Multi-scale Gaussian representation and outline-learning based cell image segmentation. BMC Bioinf. 14, 1–14.

Farhan, M., Yli-Harja, O., Niemistö, A., 2013b. A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search. Pattern Recogn. 46, 741–751.

Freeman, H., 1961. On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput. EC-10, 260–268.

Huang, D.C., Hung, K.D., Chan, Y.K., 2012. A computer assisted method for leukocyte nucleus segmentation and recognition in blood smear images. J. Syst. Software 85, 2104–2118.

Ko, B.C., Seo, M.S., Nam, J.Y., 2009. Microscopic cell nuclei segmentation based on adaptive attention window. J. Digit. Imaging 22, 259.

Ko, B.C., Gim, J.W., Nam, J.Y., 2011. Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake. Micron 42, 695–705.

Kong, H., Gurcan, M., Belkacemboussaid, K., 2011. Partitioning histopathological images: an integrated framework for supervised color-Texture segmentation and cell splitting. IEEE Trans. Med. Imaging 30, 1661–1677.

Kontschieder, P., Bulo, S.R., Bischof, H., Pelillo, M., 2011. Structured class-labels in random forests for semantic image labelling. International Conference on Computer Vision 2190–2197.

Kumar, S., Ong, S.H., Ranganath, S., Ong, T.C., Chew, F.T., 2006. A rule-based approach for robust clump splitting. Pattern Recogn. 39, 1088–1098.

Pan, C., Dong, S.P., Yoon, S., Yang, J.C., 2012. Leukocyte image segmentation using simulated visual attention. Expert Syst. Appl. 39, 7479–7494.

Prinyakupt, J., Pluempitiwiriyawej, C., 2015. Segmentation of white blood cells and comparison of cell morphology by linear and naïve Bayes classifiers. Biomed. Eng. Online 14, 1–19.

Rezatofighi, S.H., Soltanian-Zadeh, H., 2011. Automatic recognition of five types of white blood cells in peripheral blood. Comput. Med. Imaging Graphics 35, 333–343.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention 234–241.

Ruberto, C.D., Loddo, A., Putzu, L., 2016. A leucocytes count system from blood smear images: segmentation and counting of white blood cells based on learning by sampling. Mach. Vision Appl. 27 (8), 1151–1160.

Saidi, M., Bechar, M.E.A., Settouti, N., Chikh, M.A., 2016. Application of pixel selection in pixel-based classification for automatic white blood cell segmentation. The Mediterranean Conference on Pattern Recognition and Artificial Intelligence 31–38.

Sertel, O., Kong, J., Catalyurek, U.V., Lozanski, G., Saltz, J.H., Gurcan, M.N., 2009. Histopathological image analysis using model-based intermediate representations and color texture: follicular lymphoma grading. J. Signal Process. Syst. 55, 169–183.

Song, Y., Cai, W., Huang, H., Wang, Y., Feng, D.D., Chen, M., 2013. Region-based progressive localization of cell nuclei in microscopic images with data adaptive modeling. BMC Bioinf. 14, 1–16.

Theera-Umpon, N., 2005. White blood cell segmentation and classification in microscopic bone marrow images. International Conference on Fuzzy Systems and Knowledge Discovery 787–796.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley, New York.

Xiaowei, C., Xiaobo, Z., Wong, S.T.C., 2006. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. IEEE Trans. Biomed. Eng. 53, 762–766.

Xing, F., Yang, L., 2016. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. IEEE Rev. Biomed. Eng. 9, 234–263.

Yan, P., Zhou, X., Shah, M., Wong, S.T.C., 2008. Automatic segmentation of high-throughput RNAi fluorescent cellular images. IEEE Trans. Inf. Technol. Biomed. 12, 109–117.

Yang, L., Meer, P., Foran, D.J., 2005. Unsupervised segmentation based on robust estimation and color active contour models. IEEE Trans. Inf. Technol. Biomed. 9, 475–486.

Yang, X., Li, H., Zhou, X., 2006. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. IEEE Trans. Circuits Syst. 53, 2405–2414.

Yang, L., Tuzel, O., Meer, P., Foran, D.J., 2008. Automatic image analysis of histopathology specimens using concave vertex graph. International Conference on Medical Image Computing & Computer-assisted Intervention 833–841.

Yang-Mao, S.F., Chan, Y.K., Chu, Y.P., 2008. Edge enhancement nucleus and cytoplast contour detector of cervical smear images. IEEE Trans. Syst. Man Cybern. 38, 353–366.

Yi, F., Chongxun, Z., Chen, P., Li, L., 2005. White blood cell image segmentation using online trained neural network. 27th Annual International Conference of the Engineering in Medicine & Biology Society 6476–6479.

Zhang, C., Xiao, X., Li, X., Chen, Y.J., Zhen, W., Chang, J., Zheng, C., Liu, Z., 2014. White blood cell segmentation by color-space-based k-means clustering. Sensors 14, 16128–16147.

Zheng, X., Wang, Y., Wang, G., Chen, Z., 2014. A novel algorithm based on visual saliency attention for localization and segmentation in rapidly-stained leukocyte images. Micron 56, 17–28.

Zhi, L., Jing, L., Xiao, X., Hui, Y., Li, X., Chang, J., Zheng, C., 2015. Segmentation of white blood cells through nucleus mark watershed operations and mean shift clustering. Sensors 15, 22561–22586.