

Retraction

Retracted: Performance Analysis of Otsu-Based Thresholding Algorithms: A Comparative Study

Journal of Sensors

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

Copyright © 2024 Journal of Sensors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] Q. Cao, L. Qingge, and P. Yang, "Performance Analysis of Otsu-Based Thresholding Algorithms: A Comparative Study," *Journal of Sensors*, vol. 2021, Article ID 4896853, 14 pages, 2021.

Research Article

Performance Analysis of Otsu-Based Thresholding Algorithms: A Comparative Study

Qinglin Cao,¹ Letu Qingge,² and Pei Yang³

¹Xining Urban Vocational & Technical College, Xining 810012, China

²Department of Computer Science, North Carolina A&T State University, Greensboro, NC 27411, USA

³Department of Computer Technology and Application, Qinghai University, Xining 810016, China

Correspondence should be addressed to Pei Yang; yangpeinmgdx@sina.com

Received 18 June 2021; Revised 12 September 2021; Accepted 8 October 2021; Published 21 October 2021

Academic Editor: Haibin Lv

Copyright © 2021 Qinglin Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image thresholding is a widely used technology for a lot of computer vision applications, and among various global thresholding algorithms, Otsu-based approaches are very popular due to their simplicity and effectiveness. While the usage of Otsu-based thresholding methods is well discussed, the performance analyses of these methods are rather limited. In this paper, we first review nine Otsu-based approaches and categorize them based on their objective functions, preprocessing, and postprocessing strategies. Second, we conduct several experiments to analyze the model characteristics using different scene parameters both on synthetic images and real-world cell images. We put more attention to examine the variance of foreground object and the effect of the distance between mean values of foreground and background. Third, we explore the robustness of algorithms by introducing two typical kinds of noises under different intensities and compare the running time of each method. Experimental results show that NVE, WOV, and Xing's methods are more robust to the distance of mean values of foreground and background. The large foreground variance will cause a larger threshold value. Experiments on cell images show that foreground miss detection becomes serious when the intensities of foreground pixels change drastically. We conclude that almost all algorithms are significantly affected by Salt&Pepper and Gaussian noises. Interestingly, we find that ME increases almost linearly with the intensity of Salt&Pepper noise. In terms of algorithms' time cost, methods with no preprocessing and postprocessing steps have more advantages. All these findings can serve as a guideline for image thresholding when using Otsu-based thresholding approaches.

1. Introduction

Image segmentation which extracts objects of interest from background is a fundamental technology for various image processing tasks. It has been widely used in various kinds of applications including text detection [1], medical image processing [2], document binarization [3], remote sensing [4], and object detection [5]. Thresholding method is one of the most well-known segmentation algorithms due to its simplicity and effectiveness [6]. The main goal of the image thresholding is to divide the pixels of an image into several subsets by selecting some intensity values. While we can choose more than one threshold values for multilevel segmentation, in this paper, we only focus on a single threshold segmentation which separates the whole image into two

parts, and each corresponds to the background and foreground. In the past decades, there were many threshold selection methods presented including histogram shape-based methods [7–9], clustering-based methods [10–12], and entropy-based methods [13, 14]. The thresholding algorithms can be generally categorized into the global thresholding and the local thresholding. In the global thresholding, all pixels of the processed image use the same threshold value. Among all existing global thresholding algorithms, Otsu's technique [6] which was first proposed by Otsu in 1979 is still one of the most frequently used clustering-based global thresholding methods [15]. Due to simplicity and effectiveness, Otsu's technique is usually used as the preprocessing step in complex applications [16, 17]. The main idea behind the Otsu's algorithm is to maximize the

between-class variance to determine the optimal threshold value. When meeting the assumption that the gray level distributions of both the foreground and background are Gaussian distributions with equal variance, Otsu's method can produce satisfactory results on real-world images [18, 19]. However, most real-world images do not satisfy this applicable condition. Therefore, the performance of Otsu's method may degrade when the gray level histogram of the image is close to unimodal or the variances of the foreground and background are significantly different. Figure 1 shows segmentation results of Otsu's method on real cell images with a unimodal histogram and large between-class variance.

Due to the limitations mentioned above, a lot of research efforts have been put into analyzing and improving the Otsu's method. Ng [20] introduced a valley emphasis term into the objective function of Otsu's method and proposed a valley emphasis algorithm (VE) to ensure the threshold value locating at the valley of the histogram. Based on the fact that histogram valley corresponds to a low grayscale probability, Ng took the grayscale probability as the valley metric and demonstrated the effectiveness of Ng's method on some test images. However, its valley metric does not take the neighborhood information into consideration. Fan and Lei [21] modified the valley metric using smoothed grayscale probability and proposed a new method named NVE to improve Otsu's method. The valley metrics used in VE and NVE depend on the grayscale probability. To better represent the histogram valley, Xing et al. [22] proposed an improved valley emphasis method using second-order derivative-based valley metric.

Hu and Gong [23] proposed a two-stage method to further expand the application scenarios of Otsu's method. In the first stage, the method detects the peaks of the histogram close to left and right boundary and smooths the image pixels whose intensities are between the corresponding intensities of the two peaks. In the second stage, Otsu's method is applied on the preprocessed image. The experimental results showed that Hu's method can provide better segmentation results than Otsu's algorithm. To better understand the limitations of Otsu's method, Xu et al. [24] studied its characteristics and proposed a two-round Otsu's algorithm. The main conclusion of Xu's work is the threshold value computed using Otsu's method tends to bias toward the class with a larger within-class variance. Inspired by Xu's work, Yuan et al. [25] introduced a weighted object variance (WOV) parameter to the Otsu's objective function and proposed an improved Otsu's method for detection. Also motivated by Xu's work, Yang et al. [26] analyzed the relationship between pixel intensity and cumulative pixel number and gave a postadjusting strategy for the threshold value acquired by Otsu's method. Not only the improved one dimensional Otsu's methods were developed which mentioned above but also many two-dimensional Otsu's algorithms have been proposed [27–30]. Cao et al. [31] claimed that the existing improved Otsu's methods could not process images with a broad histogram or a flat valley. Many of the existing methods, especially two-dimensional Otsu's methods, are not parameter free. To overcome this

challenge, Cao et al. [31] proposed a novel parameter-free method which can achieve a more accurate and robust performance. In addition, there are also studies trying to improve the performance of Otsu's method in the case of multiple thresholds [32]. A typical way is to use bionic algorithms (e.g., artificial bee colony algorithm [32, 33]) to search optimal multiple thresholds instead of using brute force methods. In this paper, we are mainly focusing on the bilevel image thresholding, and we will not further review the multilevel thresholding methods.

While a large number of improved Otsu's methods have been proposed, there are a few studies focusing on the performance analysis of the algorithms in-depth. Goh et al. [34] studied the performance of Otsu's technique using Monte Carlo statistical method. However, the performance comparison among different Otsu-based algorithms is still understudied, and also the test images used in existing studies are significantly different. To the best of our knowledge, there is no existing study on the categorization strategy of a various Otsu-based algorithms. In this paper, we first give a categorization method for different Otsu-based methods based on the objective functions, preprocessing, and post-processing strategies. Second, a real-world image dataset and a Monte Carlo-based image synthetic method are employed for the evaluation of Otsu-based thresholding algorithms. Furthermore, several typical corruptions are adopted to test different algorithms. The main contributions of the paper are as follows:

- (1) We propose a categorization method for different improved Otsu-based algorithms based on their objective functions, preprocessing, and postprocessing strategies
- (2) The characteristics of each Otsu-based method are analyzed in-depth. More specifically, the effect of variance and distance between mean values of foreground and background, the ratio of foreground object in the whole image are discussed using a Monte Carlo-based image synthetic method
- (3) The robustness of each Otsu-based method to two typical kinds of corruptions is studied on a real-world cell image dataset, and the performance of each algorithm corresponding to different noise intensities is analyzed
- (4) We compare the time cost of all test algorithms and find that preprocessing or postprocessing steps may significantly increase algorithm's consuming time when implementing by Matlab if the preprocessing or postprocessing does not meet the language characteristics

The remainder of the paper is organized as follows. Section 2 introduces the basic principle of Otsu's method and reviews the recent improved algorithms based on the Otsu's method using a categorization strategy. Section 3 presents the characteristics and performance analytics for the compared Otsu-based algorithms by conducting experiments on the synthetic images and real-world cell images. Section 4 presents the discussion and conclusion.

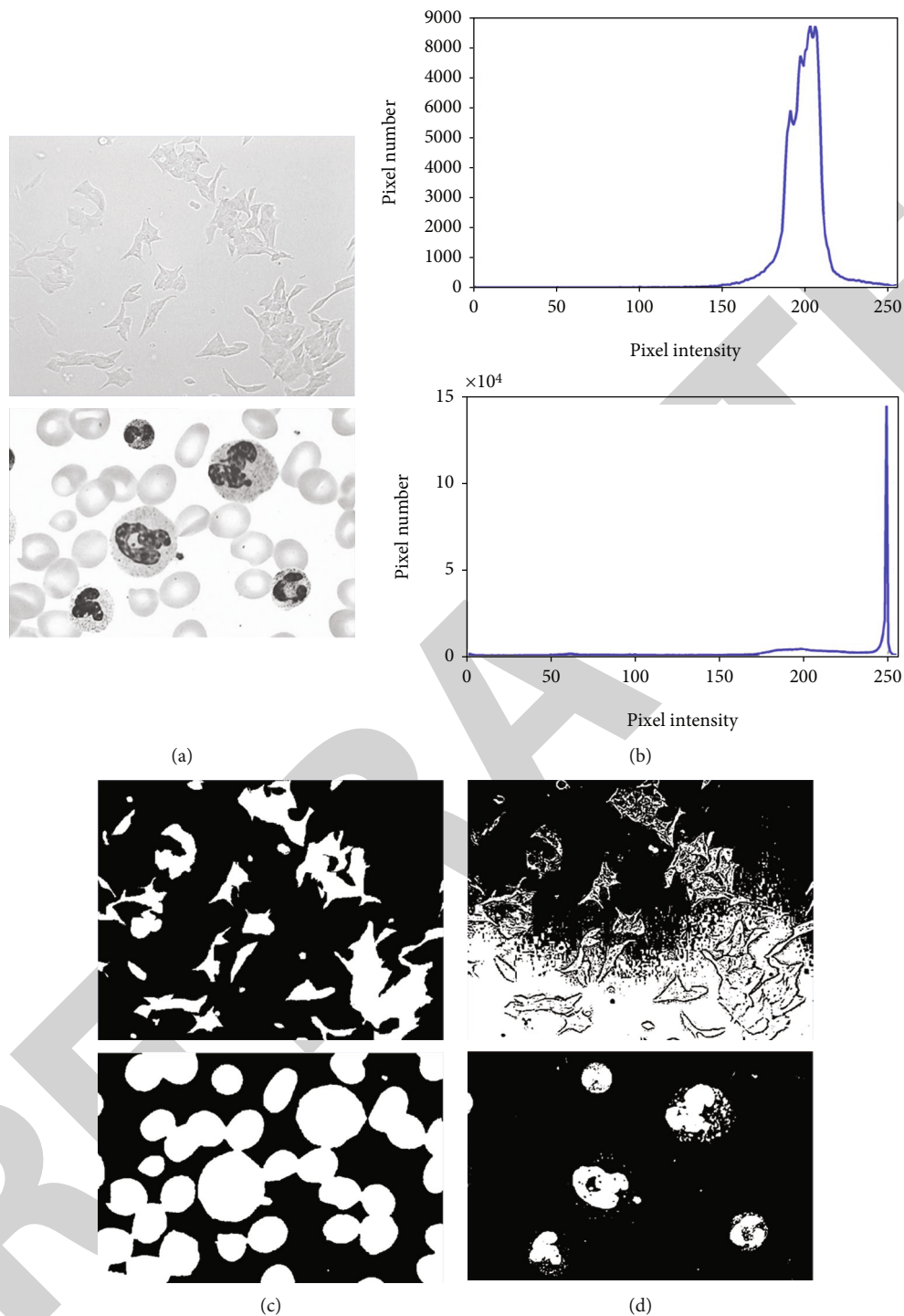


FIGURE 1: Segmentation results of real-world images which do not meet the applicable conditions of Otsu’s method. (a) Real-world graylevel images. (b) Corresponding histograms. (c) Manually labeled ground truths. (d) Segmentation results of Otsu’s method.

2. Overview of Otsu-Based Thresholding Methods

Otsu’s method is one of the most frequently used automatic thresholding algorithms. In this section, we introduce the basic principle of Otsu’s algorithm and give a

brief review of recently improved Otsu-based thresholding methods.

Suppose I is a grayscale image of size $m * n$, and the intensities of pixels of I are ranging from 0 to $L - 1$. All pixels of I can be divided into two sets C_0 and C_1 when giving a specific intensity value t as a threshold. If the mean

TABLE 1: A description of each Otsu-based algorithm discussed in this paper.

Method	Objective function	Preprocessing	Postprocessing
Otsu's [6]	$\sigma(t)$	—	—
VE [20]	$\sigma(t) \cdot w_v(t)$	—	—
NVE [21]	$\sigma(t) \cdot w_v(t)$	—	—
Xing's [22]	$\sigma(t) \cdot w_v(t)$	—	—
WOV [25]	$P_0(t) \cdot P_0(t) \cdot (\mu_0(t) - \mu(t))^2 + P_1(t) \cdot (\mu_1(t) - \mu(t))^2$	—	—
Cao's [31]	$P_0(t) \cdot P_1(t) \cdot ((\mu_0(t) - \mu_1(t))^2 + (\mu_0(t) - \mu(t))^2 + (\mu_1(t) - \mu(t))^2)$	—	—
Hu's [23]	$\sigma(t)$	√	—
Xu's [24]	$\sigma(t)$	√	—
Yang's [26]	$\sigma(t)$	—	√

intensity of C_0 and C_1 is denoted as $\mu_0(t)$ and $\mu_1(t)$, and the probability of C_0 and C_1 are denoted as $P_0(t)$ and $P_1(t)$, the between class variance of Otsu's method can be defined as:

$$P_0(t) = \sum_{i=0}^t p_i, \quad (1)$$

$$P_1(t) = \sum_{i=t+1}^{L-1} p_i = 1 - P_0(t), \quad (2)$$

$$\sigma(t) = P_0(t) \cdot (\mu_0(t) - \mu)^2 + P_1(t) \cdot (\mu_1(t) - \mu)^2, \quad (3)$$

where p_i is the probability of intensity i , which can be described as $p_i = f(i)/(m * n)$ if $f(i)$ represents the number of pixels with intensity i . μ is the mean intensity of the whole image I , which can be expressed as $\mu = P_0(t) \cdot \mu_0(t) + P_1(t) \cdot \mu_1(t)$.

With the above definitions, we can compute the optimal threshold t^* of Otsu's method by solving the following optimization problem:

$$t^* = \arg \max_t \sigma(t) = \arg \max_t P_0(t) \cdot (\mu_0(t) - \mu)^2 + P_1(t) \cdot (\mu_1(t) - \mu)^2. \quad (4)$$

The objective function described in equation (4) indicates that the best threshold of Otsu's method maximizes the weighted sum of the square of distance between mean intensity of foreground and the whole image and the square of distance between mean intensity of background and the whole image, which is also known as the interclass variance.

There are two main improvement strategies for the original Otsu's method. One is to modify the objective function to make the threshold value more reasonable, and the other is to add preprocessing and/or postprocessing steps. Table 1 lists algorithms and their corresponding objective functions for the performance comparison purpose in this paper. Following the two main improvement strategies, we can roughly divide the eight methods in Table 1 into two categories. Methods including VE [20], NVE [21], Xing's, WO, and Cao's are only focusing on improving the objective

function, and Hu's, Xu's, and Yang's methods introduce preprocessing and/or postprocessing steps. In the following, we will give a brief introduction to the other eight Otsu-based algorithms.

As described in Table 1, the first improved algorithm VE modified the objective function of Otsu's method by adding a new valley metric term $w_v(t)$ to ensure the selected threshold value is more likely located at the valley of the histogram. The valley metric used in VE algorithm is defined as below:

$$w_v(t) = 1 - p_t, \quad (5)$$

where p_t represents the probability of occurrence of pixels with intensity t . The valley metric does not take neighborhood information into consideration, and in NVE the metric is modified as:

$$w_v(t) = 1 - \bar{p}_t, \quad (6)$$

where $\bar{p}_t = \sum_{i=-m}^m p_{t+i}$ and the value of m is suggested to be 5 according to Fan and Lei's study [21].

A new second-order derivative based valley metric was proposed by Xing et al. [22] in which the valley metric proposed was defined as equation (7) below:

$$w_v(x) = \frac{\Delta^2 p_x - \min(\Delta^2 p_{(\cdot)})}{\max(\Delta^2 p_{(\cdot)}) - \min(\Delta^2 p_{(\cdot)})}, \quad (7)$$

where $p_{(\cdot)}$ represents the probability of all useable gray levels.

Besides the three valley emphasis methods above, Cao's method and WO algorithm also belong to the first category which are focusing on modifying the objective function. In WO, the influence of the foreground variance is increased, and the weight of the first term of Otsu's method is changed from $P_0(t)$ to $P_0(t) \cdot P_0(t)$. To maximize the distance between mean values of foreground and background, Cao et al. added a distance term into the objective function, which is defined as:

$$(\mu_0(t) - \mu_1(t))^2. \quad (8)$$

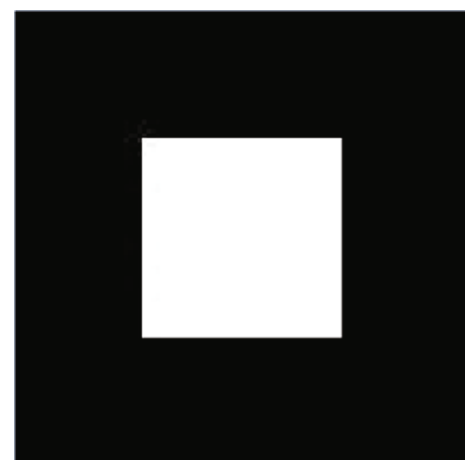
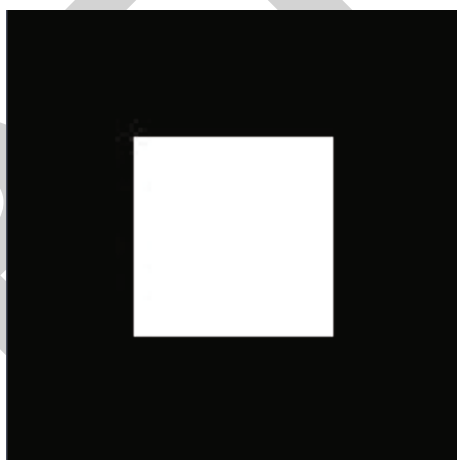
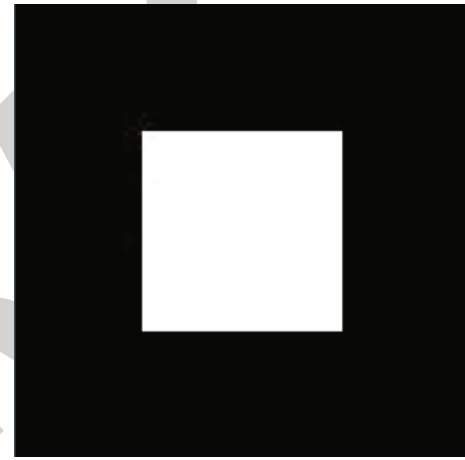
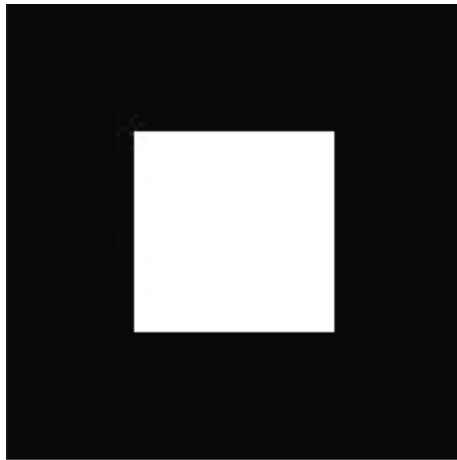
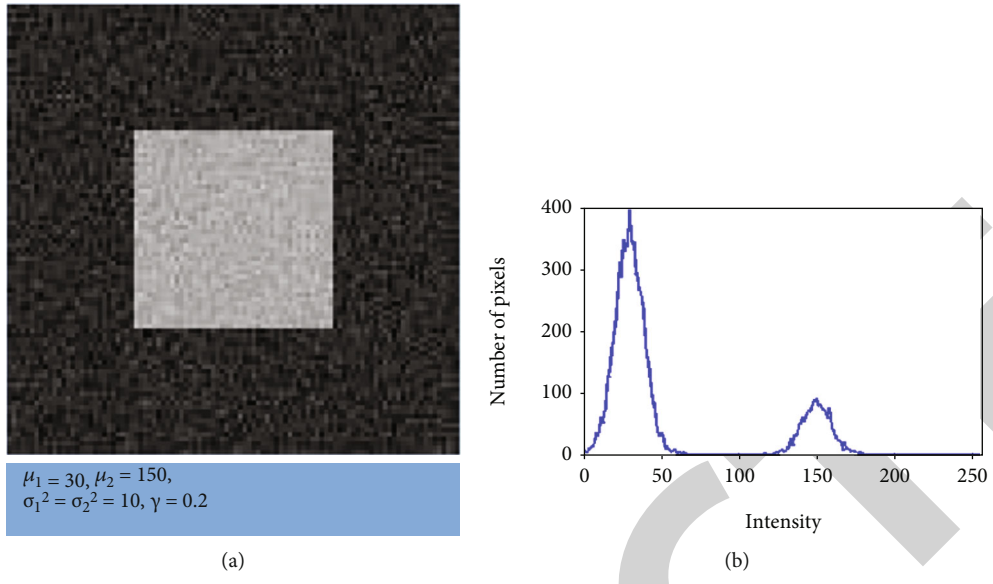


FIGURE 2: Continued.

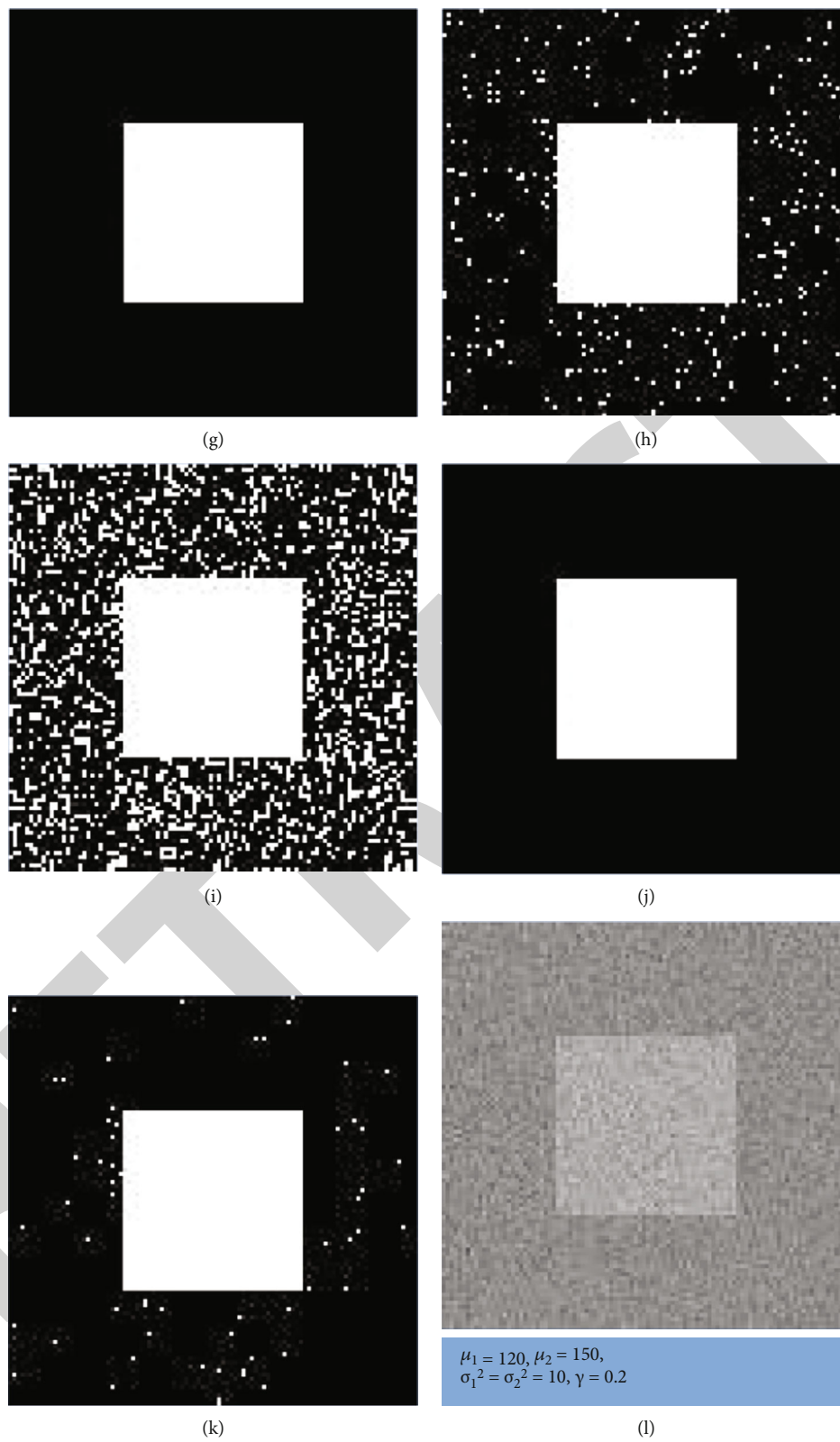


FIGURE 2: Continued.

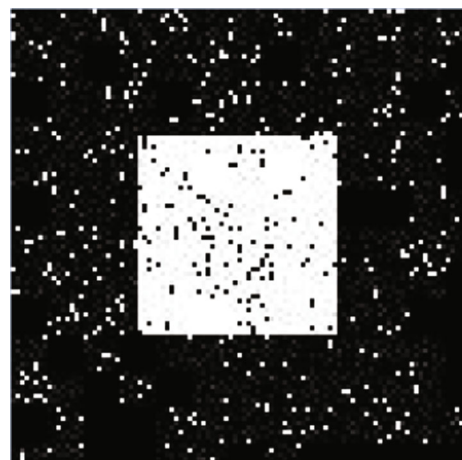
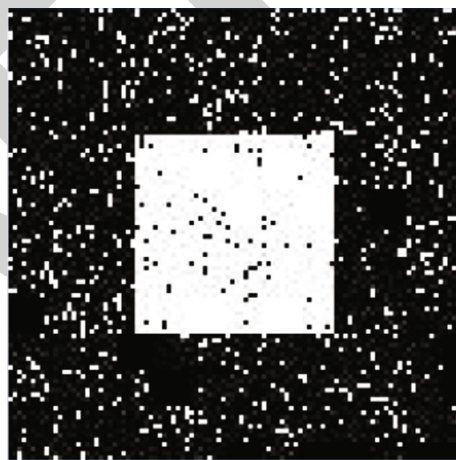
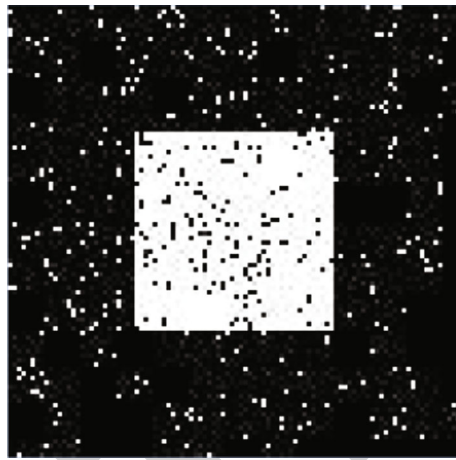
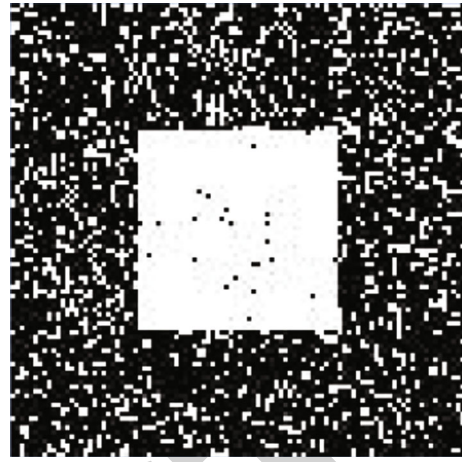
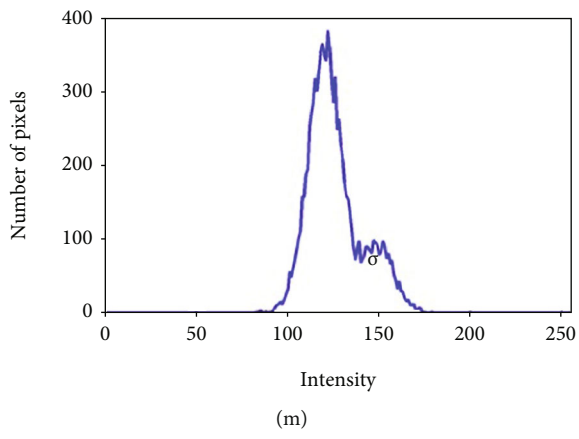


FIGURE 2: Continued.

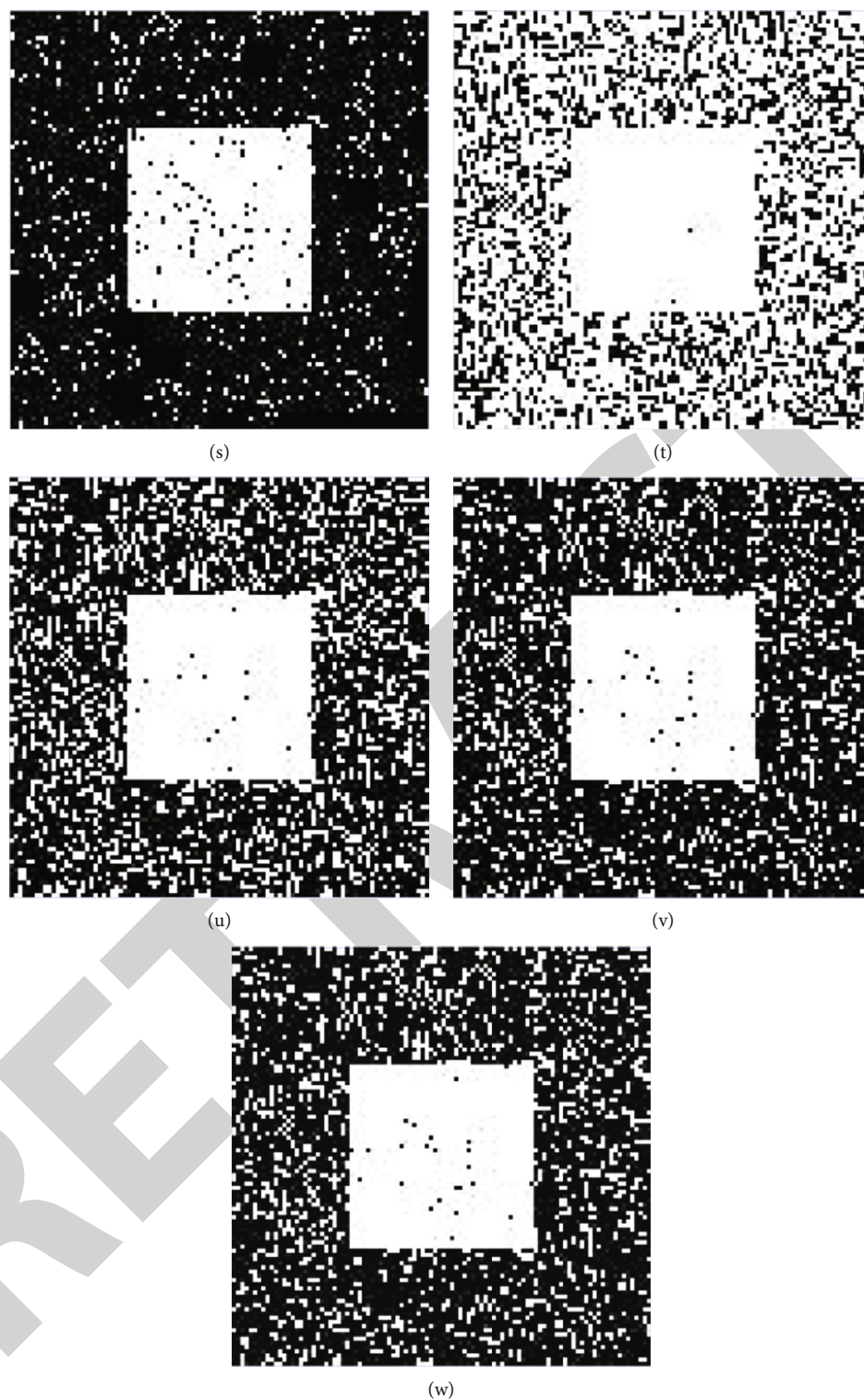


FIGURE 2: Segmentation results of each algorithm on synthetic images. (a) and (l) are synthetic images using different parameters; (b) and (m) are histograms of (a) and (l), respectively; (c) and (n) are results of Otsu's method; (d) and (o) are results of Cao's method; (e) and (p) are results of VE; (f) and (q) are results of NVE; (g) and (r) are results of WOV; (h) and (s) are results of Xing's method; (i) and (u) are results of Xu's method; (j) and (v) are results of Hu's method; (k) and (w) are results of Yang's method.

The second category is based on adding preprocessing and/or postprocessing steps. In Hu's study, a preprocessing scheme is applied on the processed gray level image

$$I(x, y) = \begin{cases} 0.25 * (|I(x-1, y) - I(x, y)| + |I(x+1, y) - I(x, y)| + |I(x, y-1) - I(x, y)| + |I(x, y+1) - I(x, y)|) + I(x, y) & I(x, y) > T_0 \text{ and } I(x, y) < T_1 \\ I(x, y) & \text{others} \end{cases} \quad (9)$$

where $I(x, y)$ represents the pixel intensity at (x, y) , T_0 and T_1 are gray levels corresponding to the first and last peaks of the histogram. The main idea of formula (9) is to take neighborhood information into consideration for pixels whose intensities belong to (T_0, T_1) , and then the Otsu's method is implemented on the smoothed image.

Different from Hu's preprocessing strategy, Xu et al. proposed a two-round Otsu's method which applied a preprocessing step on intensity probability of the processed gray image. Xu et al. assumed that the pixel intensity of foreground in an image is larger than background, the interclass variance of foreground is significantly large, and the real optimal threshold value should be smaller than Otsu's result. To make the threshold value sounder, Xu et al. proposed to apply the following preprocessing strategy on pixel intensity probability and calculate the optimal threshold value using Otsu's algorithm based on new pixel intensity probabilities in.

$$\begin{aligned} P_0(t) &= \frac{\sum_{i=0}^t P_i}{\sum_{i=0}^T P_i}, P_1(t) = \frac{\sum_{i=t+1}^T P_i}{\sum_{i=0}^T P_i}, \\ \mu_0(t) &= \frac{\sum_{i=0}^t i * P_i}{\sum_{i=0}^T P_i}, \mu_1(t) = \frac{\sum_{i=t+1}^T i * P_i}{\sum_{i=0}^T P_i}. \end{aligned} \quad (10)$$

Yang et al. also proposed a postprocessing strategy for threshold tuning based on the relationship between $P(T)$ and $P(t)$ defined as below:

$$\begin{cases} P(T) = \sum_{i=0}^T p_i, P(t) = \sum_{i=0}^t p_i & \mu < T, \\ P(T) = \sum_{i=T+1}^{L-1} p_i, P(t) = \sum_{i=t}^{L-1} p_i & \mu \geq T, \end{cases} \quad (11)$$

where T represents the threshold value obtained by Otsu's method, L and μ are gray levels and mean intensity of the whole image defined as above. For a given threshold value β , the new optimal threshold value can be described as the solution of the following optimization problem.

$$t^* = \arg \min_t \left| \frac{P(t)}{P(T)} - \beta \right|. \quad (12)$$

before implementing standard Otsu's method. The image preprocessing can be described as:

As we discussed, two different improvement ideas are employed in the eight improved Otsu's methods. The first category of improved Otsu's methods is aimed at forming a more suitable objection function by adding various kinds of constraint components, which provides some helpful clues for the optimal threshold value searching. Another kind of improved algorithms does preprocessing or postprocessing either on the input image or on its histogram instead of directly modifying the objection function, which, as a matter of fact, is a hybrid technology. In the following sections, experiments will be conducted to compare the performance of each algorithm introduced above.

3. Performance Analysis of Otsu-Based Thresholding Methods

In this section, the performance of each Otsu-based algorithm will be evaluated. All experiments are implemented using Matlab R2012b on a PC with Windows 7. The experiments are conducted on two kinds of images which are synthetic images and real world images. The following influenced factors will be discussed on synthetic images and real world images:

- (1) *Variance and Distance between Mean Values of Foreground and Background.* We will generate different synthetic images whose pixel intensity distributions of foreground and background are approximately normal distribution, and the effect of variance and distance between mean values will be discussed
- (2) *The Ratio of Foreground Object in the Whole Image.* The foreground object size is another important factor that will influence the thresholding results. Images with different foreground ratios will be generated and used to study the effect of foreground ratio on segmentation results
- (3) *Image corruption noises including Salt&Pepper noise and Gaussian noise are applied to test the robustness of each algorithm*

3.1. Testing on Synthetic Images. In this section, we will study the relationship between algorithms' performance and variances and distance of mean values of foreground

and background. The segmentation ratio measurement is adopted as the evaluation metric [34], which is defined as:

$$\zeta = \frac{\text{no.of segmented object pixel}}{\text{no.of true object pixel}}. \quad (13)$$

From the definition, it is obvious that if the segmentation is completely correct, the segmentation ratio will be 1, other values are corresponding to under segmentation ($\zeta < 1$) and oversegmentation ($\zeta > 1$), respectively.

To study the first two items of the above three affecting factors, we generate appropriate images using different parameters. The intensities of foreground and background of the generated images are all approximately normally distributed as follows:

$$\begin{cases} I(x, y) \sim N(\mu_1, \sigma_1^2), I(x, y) \in \text{background}, \\ I(x, y) \sim N(\mu_2, \sigma_2^2), I(x, y) \in \text{foreground}, \end{cases} \quad (14)$$

where μ_1, μ_2 are mean values of background and foreground, and σ_1^2, σ_2^2 denote their respective variances. For an image with m rows and n columns, we denote the ratio of foreground object in the whole image as γ , which is defined as:

$$\gamma = \frac{\text{no.of foreground pixel}}{m * n}. \quad (15)$$

Figure 2 shows the binarization results of each algorithm on two synthetic images with different parameters. The foreground ratios of the two images are both 20%, and the variances of foreground and background are fixed as 10. The first synthetic image is almost bimodal, and the foreground and background distributions of the second synthetic image are significantly overlapped. From Figure 2, we can see that most methods obtain good segmentation results on the first synthetic image except Xing's, Xu's, and Yang's. However, the segmentation results become worse on the second synthetic image. The segmentation results indicate that the distance of mean values of foreground and background will significantly affect the algorithms' performance, and more details will be discussed next.

Figure 3 demonstrates the relationship between ζ and η which is defined as formula (16), where η represents the distance between mean values of foreground and background. The trend is obvious that a small η will result in a larger segmentation ratio and lead to a more serious oversegmentation as shown in Figure 2. Combining the results of Figures 2 and 3, we conclude that distance between foreground and background is an important factor that affects the segmentation results. Among all the compared algorithms, Xing's method seems more robust, and VE is the most influenced scheme.

$$\eta = \mu_2 - \mu_1. \quad (16)$$

To test the influence of foreground object size, we further conduct experiments to verify the performance of algorithms on synthetic images with different foreground ratio γ

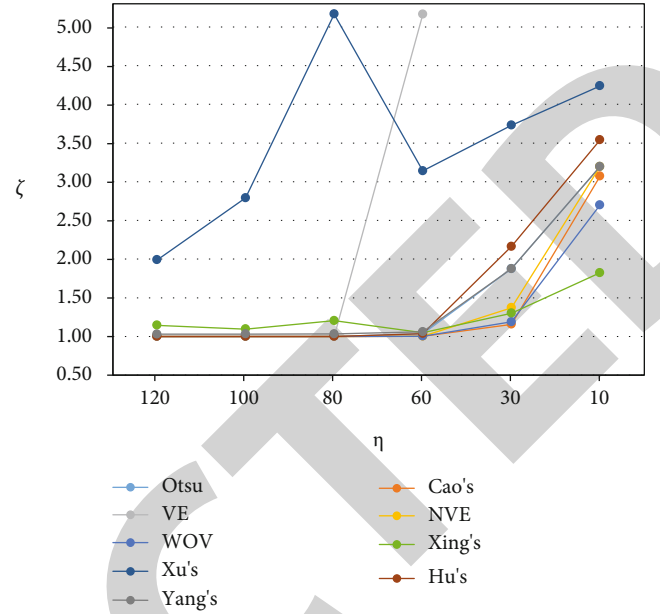


FIGURE 3: Relationship between segmentation ratio and distance between mean values of foreground and background.

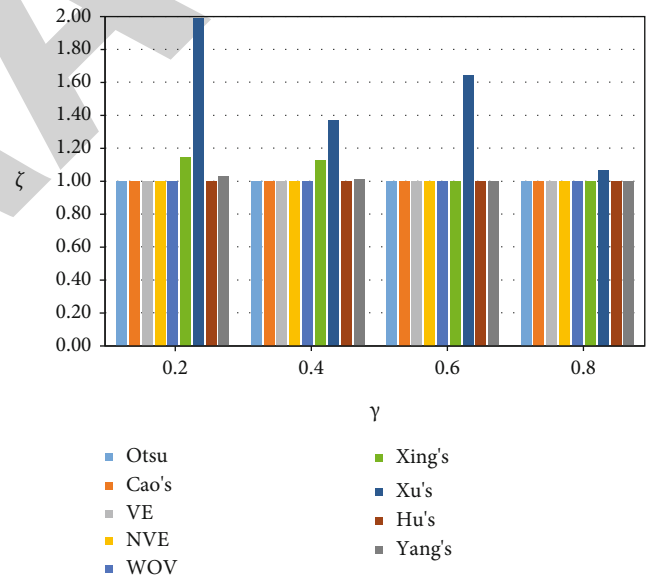


FIGURE 4: Segmentation ratio of each algorithm under different foreground ratio.

defined as formula (13). Through this experiment, we fix $\mu_1 = 30, \mu_2 = 150$ and $\sigma_1^2 = \sigma_2^2 = 10$. Figure 4 shows the segmentation ratio of each algorithm on four synthetic images with foreground ratio ranging from 0.2 to 0.8 with step length of 0.2. From Figure 4, we can see that the segmentation ratio of most algorithms except Xing's and Xu's methods are stable, which means the foreground object ratio has little influence on the segmentation results. Xing's and Xu's methods tend to oversegment according to the value of ζ (where $\zeta > 1$ indicates oversegmentation as described above). However, the foreground seems to be properly

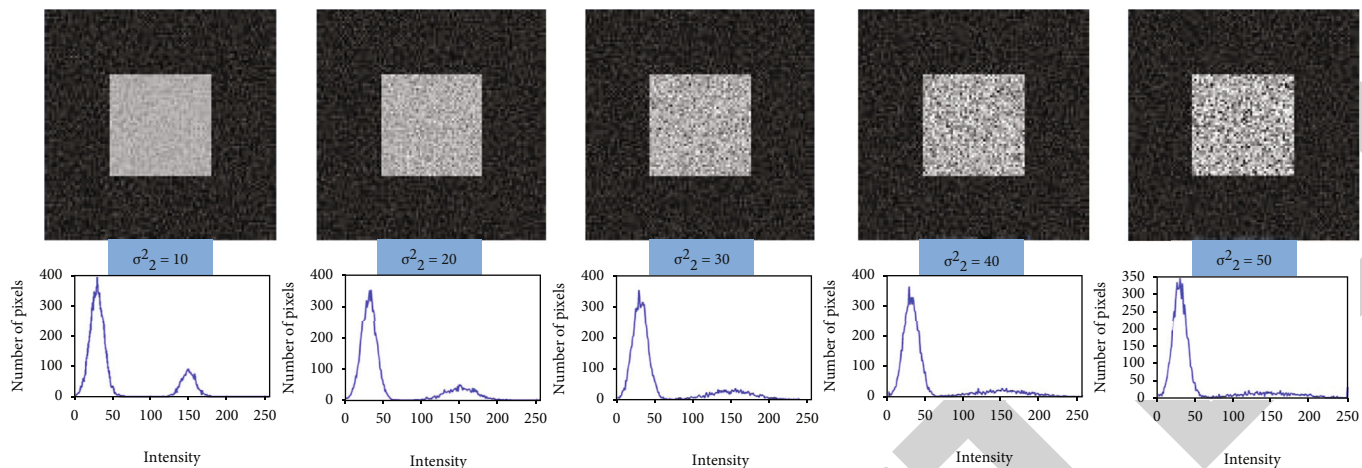


FIGURE 5: Synthetic images with different foreground variances, the first row is synthetic images, and second row is corresponding histograms.

TABLE 2: Threshold values of each algorithm on different synthetic images.

σ_2^2	Otsu	Cao's	VE	NVE	WOV	Xing's	Xu's	Hu's	Yang's
10	92	91	90	92	91	46	36	90	53
20	93	97	88	89	94	45	36	90	54
30	94	98	89	90	94	119	37	90	65
40	96	101	96	93	97	44	38	90	85
50	99	106	96	96	100	58	39	90	89

segmented when γ is large enough, which may indicate that a larger foreground ratio seems to reduce the difficulty of the segmentation task.

In this part, synthetic images with different foreground variances are generated and tested to verify the influence of different foreground variances. The mean values of foreground and background are set as $\mu_1 = 30, \mu_2 = 150$, respectively, and the background variance is fixed as $\sigma_1^2 = 10$. The generated images and their corresponding histograms are shown in Figure 5. As introduced above, the part of histogram corresponding to the background of each synthetic image is extremely similar to each other, and the part corresponding to the foreground becomes more flat as σ_2^2 increases from 10 to 50.

Table 2 illustrates the threshold values of each algorithm on synthetic images as shown in Figure 5. From Table 2, we find that the threshold value increases as the foreground variance increases for most algorithms, which is consistent with the conclusion in Xu's work that the threshold value tends to be close to the class with larger variance.

Figure 6 demonstrates the segmentation ratio of each method under different foreground variances on synthetic images. Although the foreground variance can influence the threshold value, it seems the segmentation ratio is not affected for most algorithms. One reason could be the distance between the mean values of foreground and background is large enough.

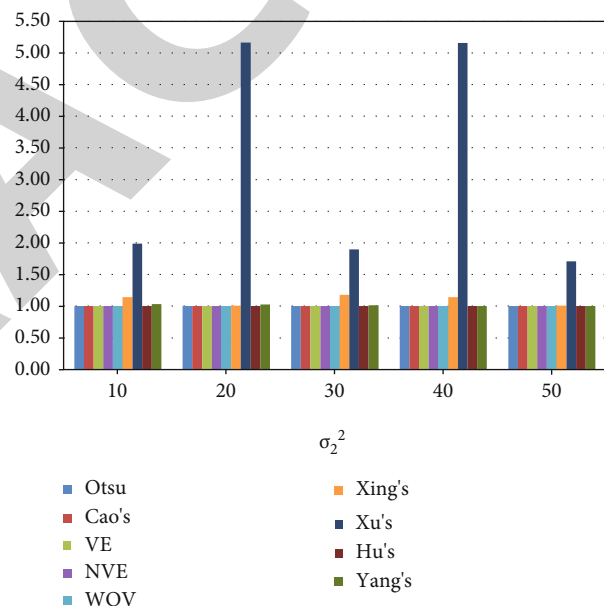


FIGURE 6: Segmentation ratio of each algorithm under different foreground variance.

TABLE 3: Average ME of each method on all cell images.

Method	ME	Method	ME	Method	ME
Otsu	0.1639	Cao's	0.1639	VE	0.1611
NVE	0.1626	WOV	0.2165	Xing's	0.1457
Xu's	0.2492	Hu's	0.1659	Yang's	0.1612

3.2. *Testing on Real-World Images.* In this section, common corruption factors are tested on a real world cell image dataset. The dataset is proposed by Xing et al. [22], which contains 22 cell images with manually labeled ground truth. Besides qualitatively evaluation, misclassification error (ME)

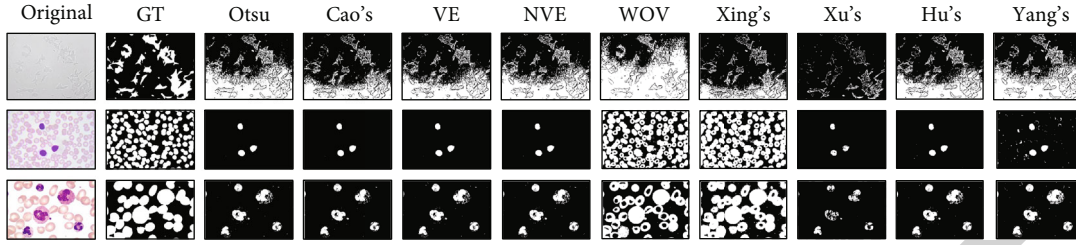


FIGURE 7: Segmentation results of each algorithm on three cell images.

is adopted as a quantitative evaluation metric which is defined as follows:

$$ME = 1 - \frac{|F_o \cap F_T| + |B_o \cap B_T|}{|B_T| + |F_T|}, \quad (17)$$

where F_o and B_o are pixel sets of foreground and background segmented by automatic thresholding method, and F_T, B_T are manually labeled foreground and background pixel sets which serve as a ground truth.

Table 3 shows the average ME values of each algorithm on 22 cell images, from which we find that Xing's method can reach the best average ME value. The advantage of Xing's method on cell images can also be found in Figure 7. Xing's method and WOV get absolute better segmentation results than other methods, especially on the second and third cell images. When the intensities of foreground object pixels change drastically as in the second and third cell images in Figure 7, foreground miss detection of most algorithms may become more serious.

Next, two kinds of noises are tested. First one is Salt&Pepper noise which is one of the most common noises. We test the influence of Salt&Pepper noise on each method on cell images. The ME values are calculated on original images and noise corrupted images with the noise intensity ranging from 0.1 to 0.7, and the relationship between average ME value of each algorithm and noise intensity can be found in Figure 8. It is obvious that all algorithms are significantly affected by Salt&Pepper noise, and the ME value grows almost linearly with increasing noise intensity δ . Since Salt&Pepper noise can be effectively removed by using median filter, it is necessary to add a filtering step before applying thresholding algorithms.

Gaussian noise is another type of common noise. In this study, we evaluate the performance of all methods on Gaussian corrupted cell images. The mean value of Gaussian is always set to 0 in this section, and the variance changes from 0 to 0.1. Figure 9 shows the relation curves between ME and δ^2 of each algorithm. Unlike Salt&Pepper noise, the ME values grow quickly if we add Gaussian noise with a small variance. However, the influence of noise decreases rapidly with the increase of variance. Among all methods, Xu's method performs best on Gaussian noise.

3.3. Algorithm Time Cost Testing. In this section, we compare the time cost of all tested algorithms. Following the analysis method introduced in Xing's study [22], it is not difficult for us to draw a conclusion that the time complexity of

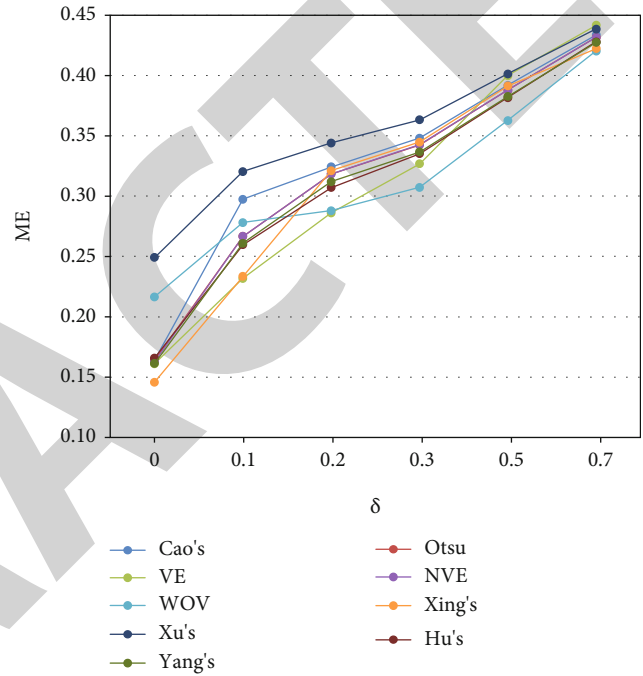


FIGURE 8: Relationship between ME and noise intensity δ .

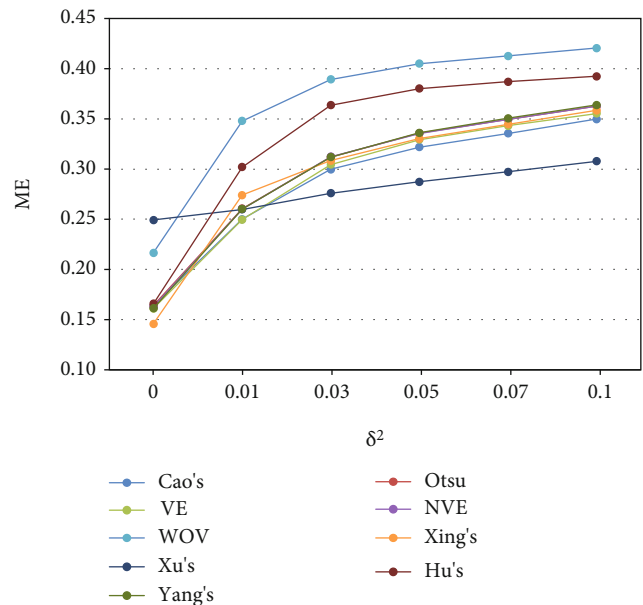


FIGURE 9: Relationship between ME and noise variance δ^2 .

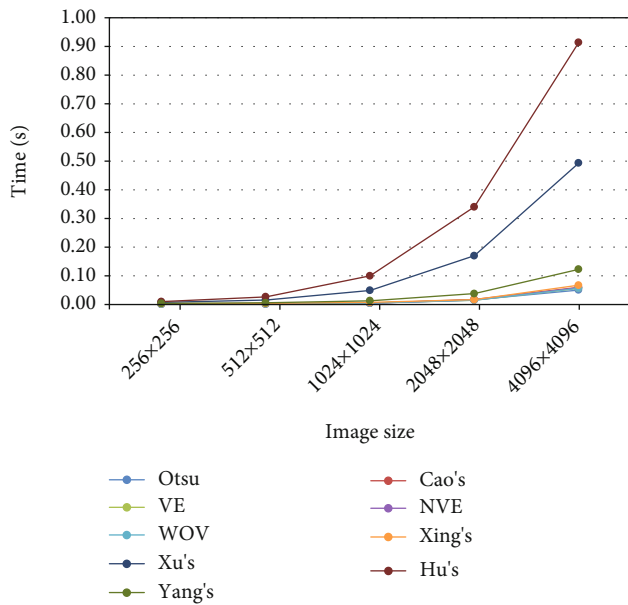


FIGURE 10: Relationship between the consuming time of each algorithm and the image size.

all test methods in this work is $O(n)$, where n represents the number of pixels of the image. However, it is important to note that we need to pay attention to the influence of the characteristics of the Matlab language on algorithm time cost. Taking Hu's method as an example, it is difficult to write the preprocessing step (defined as equation (9)) in matrix form, and this will result in an increase in algorithm's execution time. To compare the actual execution time of each algorithm, we conduct experiments using Matlab R2012b on a PC with Intel Core 2.30 GHz CPU and 8.0 GB memory.

Figure 10 shows the relationship between the consuming time of each method and the image size, and the results are the average of ten runs. It is obvious that the consuming time of methods with no preprocessing and postprocessing changes slowly with increasing image size. On the other hand, Hu's, Xu's, and Yang's methods consume more time than other methods, and this becomes increasing significant with the increase of image size. As each algorithm has the same time complexity, the significant difference in time cost is related to whether the implementation code meets the language characteristics of Matlab.

4. Conclusion

In this study, the performance of nine Otsu-based thresholding algorithms is analyzed. Experiments on synthetic images indicate that the distance between the mean values of foreground and background η can significantly affect the algorithms' performance. Among the discussed algorithms above, NVE, WOV, and Xing's methods are more robust to η . The variance of foreground can also affect the threshold and the segmentation result. The threshold value of almost all Otsu-based algorithms tends to increase when the foreground variance increases. Experimental results on cell

images indicate that foreground miss detection becomes serious when the intensity of foreground pixels changes drastically, and all the nine algorithms are not robust enough to both Salt&Pepper and Gaussian noises. In addition, preprocessing or postprocessing steps which do not meet the language characteristics will significantly increase algorithm's consuming time when implementing the algorithms by Matlab. These facts can serve as a guideline for Otsu-based thresholding applications in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (Grant no. 61866031) and the Science Technology Foundation for Middle-aged and Young Scientist of Qinghai University (2018-QGY6).

References

- [1] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 13-18, pp. 2963-2970, San Francisco, CA, USA, 2010.
- [2] L. Lin, W. Yang, C. Li, J. Tang, and X. Cao, "Inference with collaborative model for interactive tumor segmentation in medical image sequences," *IEEE Transactions on Cybernetics*, vol. 46, pp. 2796-2809, 2017.
- [3] S. He and L. Schomaker, "DeepOtsu: document enhancement and binarization using iterative deep learning," *Pattern Recognition*, vol. 91, pp. 379-390, 2019.
- [4] B. Han and Y. Wu, "A novel active contour model based on modified symmetric cross entropy for remote sensing river image segmentation," *Pattern Recognition*, vol. 67, pp. 396-409, 2017.
- [5] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337-1342, 2003.
- [6] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, pp. 146-168, 2004.
- [7] R. Azreal and D. Pilar, "Histogram concavity analysis as an aid in threshold selection," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 231-235, 1983.
- [8] R. Guo and S. Pandit, "Automatic threshold selection based on histogram modes and a discriminant criterion," *Machine Vision Applications*, vol. 10, no. 5-6, pp. 331-338, 1998.
- [9] M. J. Carlotto, "Histogram analysis using a scale-space approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp. 121-129, 1987.

- [10] T. W. Ridler, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 630–632, 1978.
- [11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [12] C. V. Jawahar, P. K. Biswas, and A. K. Ray, "Investigations on fuzzy thresholding based on fuzzy clustering," *Pattern Recognition*, vol. 30, no. 10, pp. 1605–1613, 1997.
- [13] H. D. Cheng, Y. Cheng, and Y. Sun, "A novel fuzzy entropy approach to image enhancement and thresholding," *Signal Processing*, vol. 75, no. 3, pp. 277–301, 1999.
- [14] P. K. Sahoo, C. Wilkins, and J. Yeager, "Threshold selection using Renyi's entropy," *Pattern Recognition*, vol. 30, no. 1, pp. 71–84, 1997.
- [15] Z. Hou, Q. Hu, and W. L. Nowinski, "On minimum variance thresholding," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1732–1743, 2006.
- [16] M. Zortea, E. S. Flores, and J. Scharcanski, "A simple weighted thresholding method for the segmentation of pigmented skin lesions in macroscopic images," *Pattern Recognition*, vol. 64, pp. 92–104, 2017.
- [17] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: a review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017.
- [18] J. Kittler and J. Illingworth, "On threshold selection using clustering criteria," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 5, pp. 652–655, 1985.
- [19] T. Kurita, N. Otsu, and N. N. Abdelmalek, "Maximum likelihood thresholding based on population mixture models," *Pattern Recognition*, vol. 25, no. 10, pp. 1231–1240, 1992.
- [20] H. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.
- [21] J. L. Fan and B. Lei, "A modified valley-emphasis method for automatic thresholding," *Pattern Recognition Letters*, vol. 33, no. 6, pp. 703–708, 2012.
- [22] G. W. Xing, P. Yang, and L. T. Qingge, "Automatic thresholding using a modified valley emphasis," *IET Image Processing*, vol. 14, no. 3, pp. 536–544, 2019.
- [23] B. Hu and N. Gong, "An improved Otsu's thresholding segmentation method," *Microelectronics and Computer*, vol. 26, pp. 153–155, 2009.
- [24] X. Y. Xu, S. Z. Xu, L. H. Jin, and E. M. Song, "Characteristic analysis of Otsu threshold and its applications," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 956–961, 2011.
- [25] X. C. Yuan, L. S. Wu, and Q. J. Peng, "An improved Otsu method using the weighted object variance for defect detection," *Applied Surface Science*, vol. 349, pp. 472–484, 2015.
- [26] P. Yang, W. Song, X. B. Zhao, R. Zheng, and L. T. Qingge, "An improved Otsu threshold segmentation algorithm," *International Journal of Computational Science and Engineering*, vol. 22, pp. 146–153, 2020.
- [27] J. Z. Liu and W. Q. Li, "The automatic thresholding of gray-level pictures via two-dimensional Otsu method," *Acta Automatica Sinica*, vol. 19, pp. 101–105, 1993.
- [28] J. L. Fan and F. Zhao, "Two-dimensional Otsu's curve thresholding segmentation method for gray-level images," *Acta Electronica Sinica*, vol. 35, pp. 751–755, 2007.
- [29] Q. Chen, L. Zhao, J. Lu, G. Kuang, N. Wang, and Y. Jiang, "Modified two-dimensional Otsu image segmentation algorithm and fast realisation," *IET Image Processing*, vol. 6, no. 4, pp. 426–433, 2012.
- [30] C. S. Sha, J. Hou, and H. X. Cui, "A robust 2D Otsu's thresholding method in image segmentation," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 339–351, 2016.
- [31] X. H. Cao, T. H. Li, H. L. Li et al., "A robust parameter-free thresholding method for image segmentation," *IEEE Access*, vol. 7, pp. 3448–3458, 2019.
- [32] H. Gao, Y. Shi, C. M. Pun, and S. Kwong, "An improved artificial bee colony algorithm with its application," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 1853–1865, 2019.
- [33] H. Gao, Z. Fu, C. M. Pun, J. Zhang, and S. Kwong, "An efficient artificial bee colony algorithm with an improved linkage identification method," *IEEE transactions on cybernetics*, pp. 1–15, 2020.
- [34] T. Y. Goh, S. N. Basah, H. Yazid, M. J. Safar, and F. S. Saad, "Performance analysis of image thresholding: Otsu technique," *Measurement*, vol. 114, pp. 298–307, 2018.