

Article

Fast Adaptive Binarization of QR Code Images for Automatic Sorting in Logistics Systems

Rongjun Chen ¹, Weijie Li ¹, Kailin Lan ¹, Jinghui Xiao ^{2,*}, Leijun Wang ¹ and Xu Lu ^{1,*}¹ School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China² Guangdong Telecommunication Plan & Design Institute Co., Ltd., Guangzhou 510665, China

* Correspondence: 13312880077@189.cn (J.X.); bruda@126.com (X.L.)

Abstract: With the development of technology, QR codes play an important role in information exchange. In order to work out the problem of uneven illumination in automatic sorting in logistics systems, an adaptive method in binarization is presented. The proposed method defines the block windows' size adaptively for local binarization based on the traits of the QR code. It takes advantage of integral images to calculate the sum of gray values in a block. The method can binarize the QR code with high quality and speed under uneven illumination. Compared with several existing algorithms, it is shown that the proposed method is more effective. The experimental results validate that the proposed method has a higher recognition accuracy and is more efficient in binarization.

Keywords: binarization; QR code; integral image; logistics

1. Introduction

Shortly, the IoT will affect many fields, such as retail, agriculture, and transportation [1]. To be more specific, people may control the heating, lighting, and other home equipment by constructing intelligent home systems using IoT technologies [2]. Moreover, the IoT may boost logistics systems' flexibility, robustness, and productivity [3,4]. While building the IoT ecosystem, the QR code, the entrance to the Internet of Things, plays a significant role. QR codes can not only store a large amount of information but also have vital error correction functions and low production costs, which makes them very valuable. In logistics systems, the packages are transmitted onto the conveyer belt. When the scanner near the conveyer belt recognizes the QR code on the packages, the local computer will upload information to the cloud and obtain a response to change the moving direction of parts of the belt to sort the packages. Then, the packages with the same destination will be put together and delivered.

It can be seen from Figure 1 that the sorting operation depends on the QR code. Combined with QR code technology, it is possible that logistics companies then implement the unmanned management of sorting and package tracking. The accuracy and speed of QR code recognition will directly affect sorting efficiency. However, in complex environments, QR code recognition is often affected by other factors, such as uneven lighting. There are several ways to deal with this factor. Wellner [5] proposed a method that used an average of surrounding pixels as a threshold to segment the image. Sauvola [6] devised a way of utilizing the average and deviation of gray values in an area to set the threshold. Zhang and Yang [7] proposed an improved algorithm based on that of Sauvola to speed up the calculation process, but the size of the processing window limited the algorithm. Di et al. [8] proposed an improved method based on Wellner's, and the results can be obtained through simple calculations. However, the algorithm is not effective in the case of severe unevenness of light. Yang and Feng [9] came up with an algorithm combining the methods of Bersen [10] and Otsu [11]. The algorithm will modify the threshold to prevent misclassification. However, several parameters in the algorithm need to be set manually,



Citation: Chen, R.; Li, W.; Lan, K.; Xiao, J.; Wang, L.; Lu, X. Fast Adaptive Binarization of QR Code Images for Automatic Sorting in Logistics Systems. *Electronics* **2023**, *12*, 286. <https://doi.org/10.3390/electronics12020286>

Academic Editor: Chiman Kwan

Received: 29 November 2022

Revised: 19 December 2022

Accepted: 27 December 2022

Published: 5 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

lacking some adaptive ability. Wu et al. [12] found a method of using BTC to segment an image into multiple blocks for binarization. This method is computationally intensive, and its effect depends on the number of segmented blocks. Chen et al. [13] proposed a window-adaptive binarization algorithm before; the proposed method has a favorable adaptive ability. The algorithm meets the requirements of the warehouse sorting systems, but it cannot meet the needs of the logistics sorting system because the automatic scoring in the logistics system has a higher requirement for efficiency.

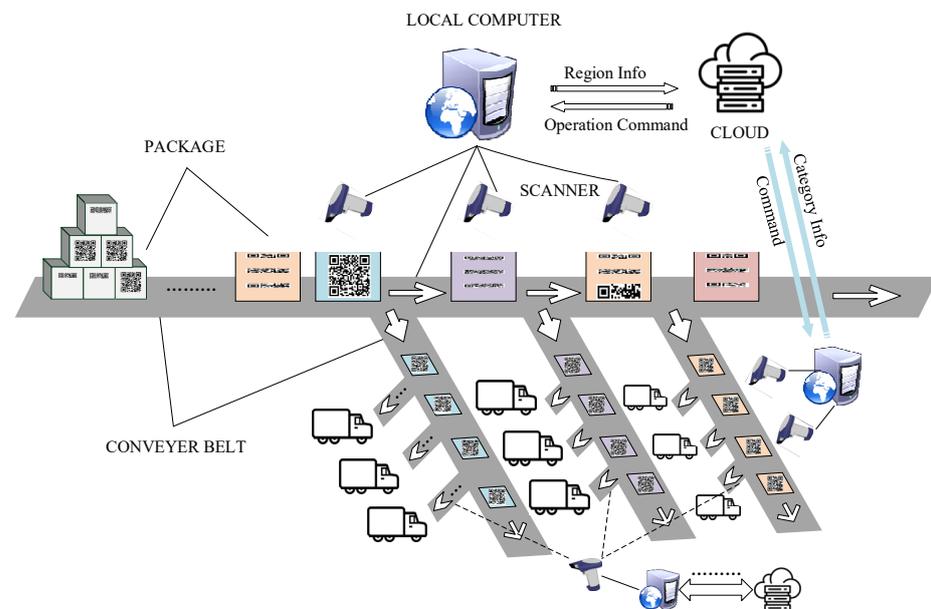


Figure 1. Workflow of automatic sorting in Logistics Systems.

In addition, feature extraction is a standard method in image processing. Ren et al. [14] presented an automatic end-to-end supervised deep learning framework to improve the accuracy of object segmentation and extraction. Sun et al. [15] proposed a novel SpaSSA approach for effective extraction and classification accuracy. Although there are substantial binarization methods in the QR code image, few feature extraction methods are utilized in QR code image processing.

Considering the needs of the logistics sorting system while extracting the features of the QR code, we propose a fast adaptive binarization algorithm, which solves the problem of difficult window definition and achieves a high recognition rating. The algorithm first determines the window size by searching for the position detection patterns and then combines the improved local binarization algorithm for processing. The experimental results show that the algorithm can not only process QR codes with different degrees of uneven illumination but also improve the quality of the fast-processed images.

2. Related Work

2.1. Features of QR Code

In the QR code, the information is encoded in a binary representation. As shown in Figure 2, the dark modules, or black modules, usually represent “1”, while the light modules, or white modules, often represent “0”. In the upper left, lower left, and upper right corners of the QR code are the position detection patterns. When the mobile device detects the three modules, it can determine the position and direction of the QR code and prepare for decoding. The three position-detection patterns in the QR code have the same size and are composed of three concentric squares. The width ratio of the dark module to the light module of the position detection pattern is 1:1:3:1:1.

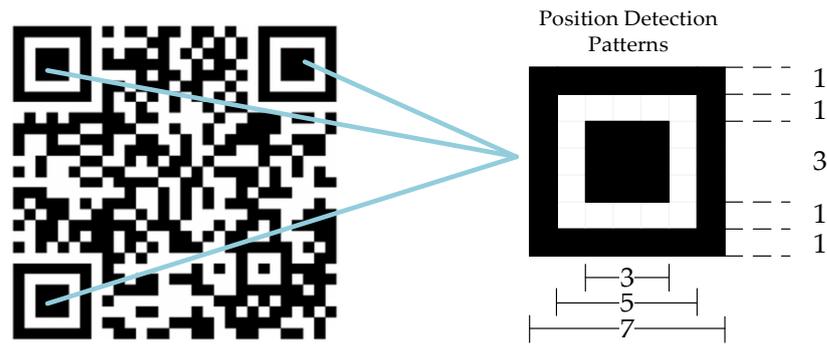


Figure 2. Position detection patterns in QR code.

2.2. Some Binarization Algorithm

At present, traditional binarization algorithms can be divided into two categories. One is the global threshold algorithm, which divides the image according to a specific gray value. The most commonly used global threshold algorithm, the Otsu algorithm, determines the threshold T by calculating the maximum between-class variance and then calculates the gray value $f(x, y)$ of each pixel on the image according to Equation (1) to obtain the final value $g(x, y)$ of the pixel. The calculation process of this kind of algorithm is relatively simple, and the calculation speed is fast. However, because the algorithm uses a uniform threshold, image details and information are seriously lost in more complex environments, and the processing effect is poor.

$$g(x, y) = \begin{cases} 255, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (1)$$

The other is a local threshold algorithm. The core of this algorithm is to use different thresholds to binarize the pixels in the image. The processed image can well restore the details of the image. Similar to the idea of Sauvola’s algorithm, first take one pixel as the center and create a local window of $w \times w$ size; second, calculate the average gray value $m(x, y)$ in the window, and then calculate the gray value standard deviation $\sigma(x, y)$ of the pixel in the window according to Equation (2); then calculate the threshold value $T(x, y)$ of the pixel according to Equation (3); afterward perform the same operation on each pixel to obtain the respective threshold value; finally, segment each pixel.

$$\sigma(x, y) = \sqrt{\frac{1}{w^2} \sum_{i=x-\frac{w}{2}}^{x+\frac{w}{2}} \sum_{j=y-\frac{w}{2}}^{y+\frac{w}{2}} (f(i, j) - m(x, y))^2} \quad (2)$$

$$T(x, y) = m(x, y) \left[1 + k \left(\frac{\sigma(x, y)}{R} - 1 \right) \right] \quad (3)$$

Although the processing effect of such algorithms is relatively sound, the amount of calculation is often too large, resulting in slow operation speed. More importantly, the algorithm is often affected by the size of the processing window. If the window definition is too small, a block phenomenon will occur. On the other hand, if the size is too large, the algorithm’s effectiveness will be reduced, and the amount of calculation will be further increased to reduce the processing speed.

Under complex lighting conditions, the traditional global binarization algorithm lacks advantages in recognition rate. At the same time, the traditional local binarization algorithm is also not ideal for processing speed and image segmentation effect. In order to accurately binarize QR codes under complex illumination, many researchers have proposed their methods. Zhou et al. [16] made improvements based on Sauvola’s method, combined with the QR code feature definition parameter R in Equation (3). However, the problem of defining the window size was not resolved. Zhang’s team [17] proposed an improved

binarization method based on background grayscale. This method combines the joint interpolation algorithm and uses the Otsu method to segment the image. Nevertheless, the effect is general in the case of multiple light sources. Yao's team [18] proposed an algorithm that fuses the improved Niblack algorithm with Otsu's algorithm. The algorithm is faster than traditional local binarization methods but is not able to binarize QR code images under different degrees of uneven illumination.

Based on the analysis of the above methods, there is still a lack of an efficient and high-quality binarization algorithm, even under complex lighting conditions. Therefore, in this paper, we bring forward an adaptive fast binarization method combined with QR code features and an integral image method.

3. Method

Most local threshold binarization algorithms have the problems of slow operation speed and obstacles in window definition. In order to solve the problems above, we conducted research based on Sauvola's algorithm. In this paper, the problem of window selection of the local threshold algorithm is solved by combining the features of the QR code image. The method in this paper first finds the position detection pattern through preliminary binarization, then obtains the window size of the local threshold algorithm. Then, it uses Sauvola's and the integral image algorithm to accelerate the threshold calculation of the QR code image and finally reconstructs the image based on the threshold to obtain the binarized image.

3.1. Preprocessing

In daily life, the quality of the unprocessed QR code image is affected by the camera quality of the mobile device, the algorithm processed after shooting, and the environment. Due to these various factors, the image will be disturbed by noise, and the image will likely add noise, so we need to perform median filtering on the QR code image to eliminate the influence of noise.

Because the window size of the algorithm in this paper depends on the central black module of the position detection pattern, we need to pre-process the QR code image to enhance the contrast to ensure the integrity of the position detection pattern after the initial binarization. Top-hat transform and histogram equalization are commonly used methods to enhance contrast.

The top-hat transform can extract the image's dark features or bright regions [19]. So using top-hat transform, the dark module in the position detection pattern can be extracted from the QR code with uneven lighting. The top-hat transformation of the grayscale image I is performed according to

$$T_{hat} = I - (I \diamond b) \quad (4)$$

where $I \diamond b$ represents the open operation and b is a structural element.

Because there is a width ratio of 1:1:3:1:1 between the modules in the position detection pattern, the size of the structure element is defined as follows:

$$b = \frac{\max(h, w)}{7} \quad (5)$$

where h is the height of image I and w is the width. After obtaining the top-hat transformed image T_{hat} , we use Otsu algorithm to perform preliminary binarization processing on the image T_{hat} . The pre-processing flowchart is shown in Figure 3.

3.2. Define Window Size Adaptively

In order to solve the problem of window size selection, we performed a series of studies. Initially, we processed the QR code image P1 (Figure 4) with a size of 300×300 and iterated from a window size of 1 pixel until the window size was 1/4 of the image width. When the window size exceeds 1/4 of the image width, this will be too time-consuming

to use in reality. Sauvola’s algorithm was used for binarizing the QR code image. After obtaining several binarized QR code images after processing, we used two objective quality evaluation standards, peak signal-to-noise ratio (PSNR), to judge the results. The larger the calculated value of the evaluation standard, the higher the image quality after binarization. Based on the evaluation of the processed pictures, we have drawn Figure 5.

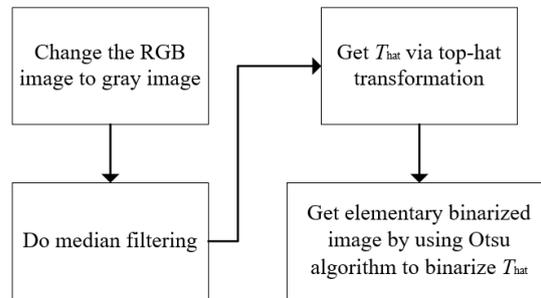


Figure 3. Flowchart of pre-processing.



Figure 4. Image P1.

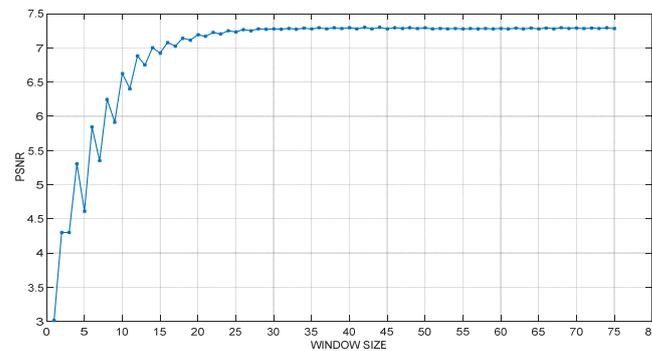


Figure 5. PSNR values after P1 processing.

From the figure, we found that when the window size is set to about 30, the values of PSNR reach the highest, after which the PSNR values remain stable. At the same time, by analyzing the QR code image in Figure 4, we found that the size of the middle black module in the position detection pattern of P1 is 30×30 , as shown in Figure 6. When the size of the window is the same as the size of the middle black block of the position detection pattern, the PSNR values reach the highest after the QR code is binarized. Finally, in order to further verify the relationship between the window size and the position detection pattern, we performed the same experiment on five 300×300 size samples (as shown in Figure 7) and obtained the results shown in Figure 8.



Figure 6. The size of middle module.

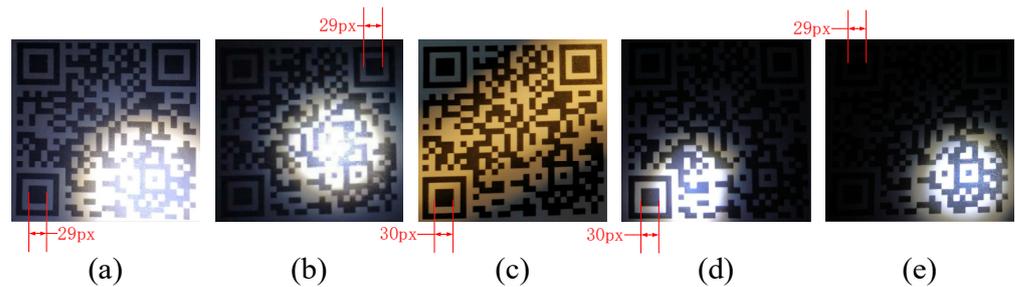


Figure 7. Experimental samples: (a) P2; (b) P3; (c) P4; (d) P5; (e) P6.

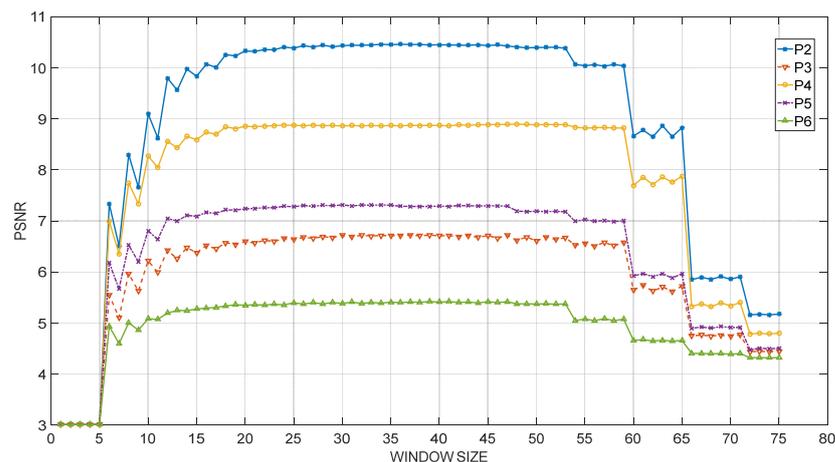


Figure 8. PSNR value curves of five 300×300 samples after processing.

As Figure 8 shows, when the window size increases, the image quality after binarization also increases. When the window size is close to the middle black module size of the position detection graphic, the binarized image quality effect reaches the best. As mentioned earlier, most local threshold algorithms currently have the problem of complicated window definitions. Therefore, the size of the black module in the center of the position detection pattern is used as the window size of the local threshold algorithm, which can effectively implement the adaptive selection of the processing window.

Subsequently, we used Adobe Photoshop to enlarge the proofs in Figure 7 to 450×450 and 900×900 , two sets of samples, and gradually increased from a window size of 1, and then binarized the samples to obtain Figure 9.

When the size of the sample is 450×450 , the middle dark module is close to 45×45 . While the dark modules are near 90×90 in the 900×900 samples. From the processing results of the two sets of samples, it can be seen that when the window is close to the dark module in the middle of the position detection pattern, the PSNR values of the image tend to be stable. However, the PSNR values tend to decline as the window size becomes extensive. For each pixel, there is a local threshold algorithm with different thresholds. When traversing each pixel to calculate the threshold, the window size will affect the calculation speed. The larger the window, the slower the calculation speed.

Therefore, the size of the dark module can be defined as the window size of the local binarization algorithm.

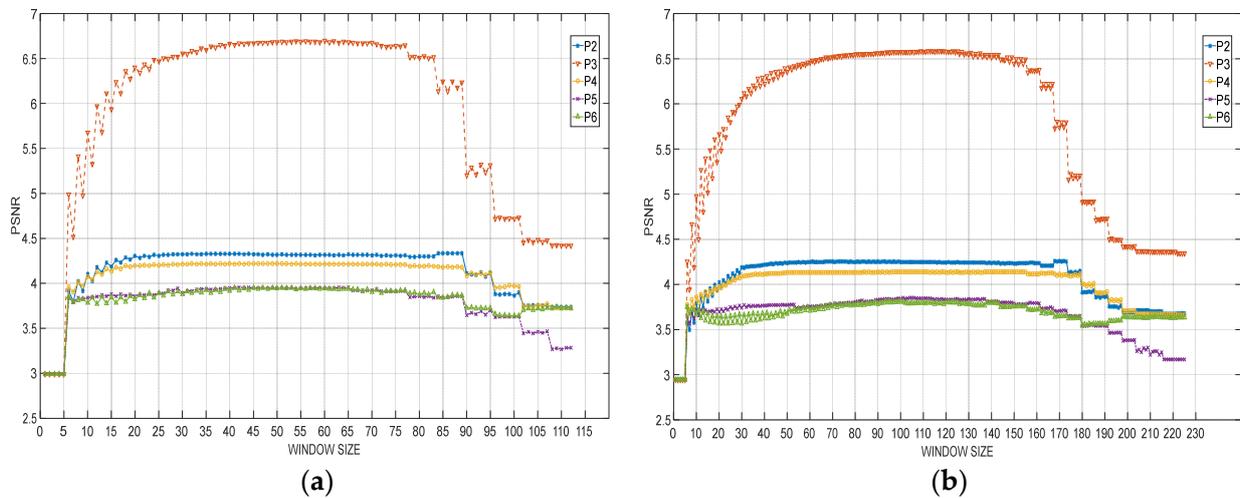


Figure 9. PSNR value curves of five samples in the same size: (a) 300 × 300; (b) 450 × 450.

As we know, an image can be viewed as a two-dimensional array. Therefore, in order to find the position detection pattern according to the characteristics of the module width ratio 1:1:3:1:1, after obtaining the preliminary binarized image, we transform the image into a one-dimensional array to obtain the array S . We can obtain the gray difference of the adjacent pixels according to

$$Gray_{diff} = S(i + 1) - S(i) \tag{6}$$

where $i \geq 1$. When $Gray_{diff} \neq 0$, we record the current value of i to obtain the index array Index_Array. Then, traverse Index_Array and find the 1:1:3:1:1 relationship. If the above proportional relationship is found, the window size can be determined according to the value of Index_Array.

3.3. Window Size Correction

It is found through experiments that in the case of slightly uneven illumination, the initial binarized image after the top-hat transformation sometimes has the situation that the position detection pattern cannot be determined. In order to improve the success rate of finding the position detection pattern, when the above method fails to obtain the window size, we try another method to obtain the window size again. Histogram equalization is an adaptive contrast enhancement tool [20]. We can use it on the grayscale image I to enhance its contrast and then perform preliminary binarization processing. Next, use the method of Section 3.2 to obtain the window size value again. If there is still no way to obtain the window size value for the histogram-equalized image, set the window size according to

$$window\ size = \frac{\min(h, w)}{10} \tag{7}$$

Finally, the method flowchart for obtaining the window size is shown in Figure 10.

We used the window size definition method just mentioned in combination with Sauvola’s algorithm to select several samples of different sizes for the binarization experiments. The original image and processing results are shown as follows.

It can be seen from Figure 11 that no matter what size the QR code image is, this window definition method is effective and can accurately segment the image.

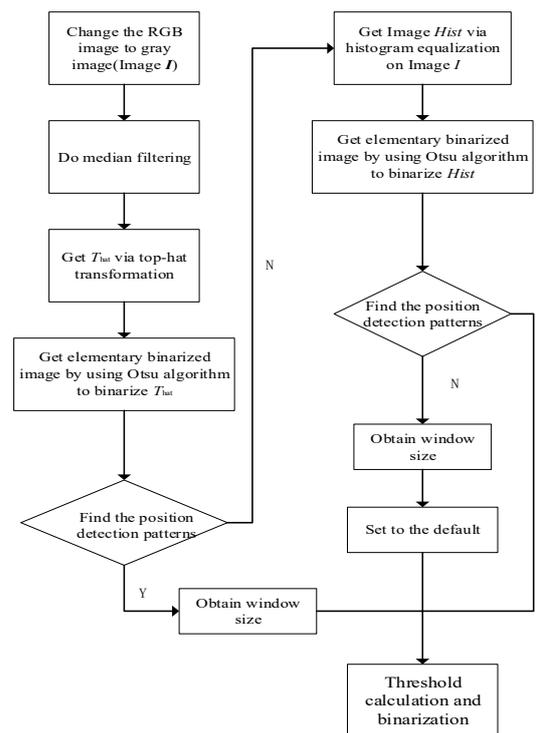


Figure 10. Flowchart of obtaining the window size.

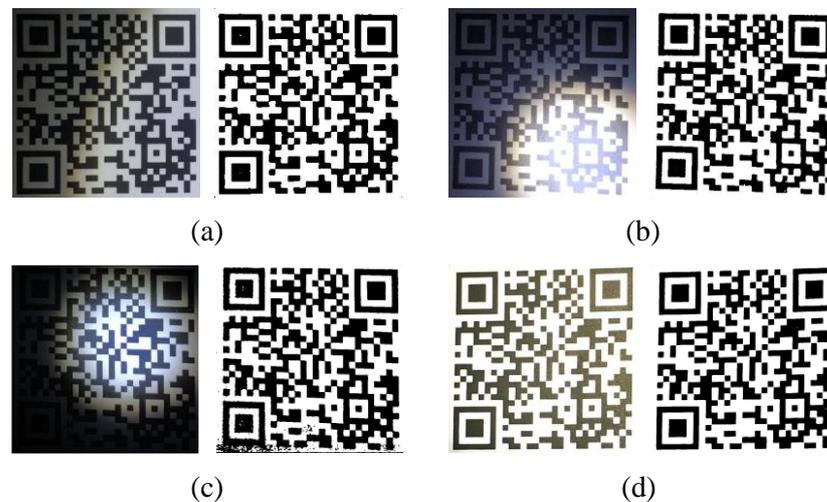


Figure 11. The original image and processing result: (a) 300 × 300; (b) 50 × 450; (c) 600 × 600; (d) 900 × 900.

3.4. Threshold Calculation and Binarization

As we all know, the operation speed of the local binarization algorithm is relatively slow. Therefore, we use the integral image algorithm to speed up the operation speed of the algorithm. The integral image is mainly used to calculate the sum of gray values in a rectangular area [21]. Therefore, during the integral image algorithm of the QR code, an integral image needs to be created—the pixels on the integral image store the corresponding total gray values in the QR code. The values on the integral image are obtained by the formula

$$Inte(x,y) = \sum_{i=1}^x \sum_{j=1}^y G(i,j) \tag{8}$$

where $G(i, j)$ represents the image T_{hat} obtained by the top-hat transformation or the image $Hist$ obtained by the histogram equalization.

Suppose there is an image A, the gray values of its image pixels are shown in Figure 12a, and the corresponding integral image is shown in Figure 12b.

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

(a)

1	3	6	10
2	6	12	20
3	9	18	30
4	12	24	40

(b)

Figure 12. The gray values of image A and its integral image: (a) image A; (b) integral image.

Generally, if you want to calculate the sum of gray values in the circled window in Figure 13a, you need to use

$$S(x, y) = \sum_{i=x-\frac{w}{2}}^{x+\frac{w}{2}} \sum_{j=y-\frac{w}{2}}^{y+\frac{w}{2}} I(x, y) \tag{9}$$

for calculation.

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

(a)

1	3	6	10
2	6	12	20
3	9	18	30
4	12	24	40

(b)

Figure 13. Pixel used to calculate the total gray value: (a) 9 pixels; (b) 4 pixels.

When using the local threshold algorithm, each time the threshold is calculated, the pixel values in the accumulation window need to be traversed, which is a time-consuming operation. In the case of using the integral image, the workload of the calculation is greatly reduced. We can just process by the formula as follows:

$$S(x, y) = Inte(x + d - 1, y + d - 1) + Inte(x - d, y - d) - Inte(x - d, y + d - 1) - Inte(x + d - 1, y - d), \tag{10}$$

where $d = w/2$. Therefore, the calculation of the sum of gray values in the window in Figure 13a only needs to be calculated as follows:

$$S(3,3) = Inte(4,4) + Inte(1,1) - Inte(1,4) - Inte(4,1) = 40 + 1 - 10 - 4 = 27, \tag{11}$$

According to Equation (11), only four calculations are needed to obtain the result, while Equation (10) requires nine calculations. Therefore, calculating the sum of the gray values using the integral image algorithm is more straightforward and faster than the original Equation (10).

Because the method in this paper uses Sauvola’s method to binarize the QR code image, when calculating the sum of the gray values in the window, combined with the integral image, the calculation amount can be effectively reduced and increase the calculation speed. Furthermore, the image can be binarized after the thresholds of all pixels are calculated using the improved Sauvola’s algorithm.

We selected 30 proofs and used two methods for experiments. One is to use Sauvola’s algorithm, and the other is to use Sauvola’s algorithm combined with the integral image algorithm. Table 1 shows the result of the experiments.

Table 1. Average consume time of the two algorithms.

Algorithm	Average Time Consuming/s
Sauvola's	4.2544
Savuola's combined with integral image	0.0157

What can be learned from Table 1 is that the time consumption of Savuola's algorithm is far greater than that of another method. It proves that using the integral image algorithm can improve binarization efficiency.

3.5. Morphological Processing

Using the localized binarized image will inevitably lead to some loss of image detail and information, sometimes causing a pseudo-boundary. We can perform an opening operation on the image to reduce the impact. The opening operation smooths the outline of the image, breaking the narrow necks and eliminating fine protrusions. According to

$$A \circ B = (A \ominus B) \oplus B \quad (12)$$

The opening operation can be understood as the structure element B is first used to corrode the image A , and then the structure element B is used to expand the result.

4. Experiments

In order to verify the effectiveness of the proposed method, we tested it in relation to image quality, recognition rate, and computation speed and used the methods in [6,8,13,18], for comparison. The parameters of the test environment are shown in Tables 2 and 3. The software used in the experiments was ZXing and WeChat. ZXing is an open-source project by Google. Most of the QR code identification software is improved based on this project. Moreover, WeChat is the most widely used QR identification software in China.

Table 2. Hardware parameters.

Hardware	Parameters
Processor	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz (8 CPUs)
Memory	16 GB SAMSUNG DDR4 2133 MHz
Smart Phone	OnePlus 3 64 GB

Table 3. Software parameters.

Software	Versions
Operating System	Windows 10 Pro 64-bit (10.0, Build 18362)
MATLAB	R2016a 64-bit
ZXing AndroidSDK	3.4.0
WeChat	7.0.10

4.1. Image Quality

The experiments used two standards, PSNR and the measure of structural similarity (SSIM), to objectively evaluate the quality of the processed image. SSIM is the same as PSNR, the more significant the value, the better the image quality. Moreover, SSIM is well-matched to perceived visual quality [22]. We conducted the experiments with several types of uneven illumination QR code images.

- (1) Figure 14 shows the effect of different algorithms in processing the QR code, which had a soft light in the lower right corner. We can still see the most details through our eyes.

- (2) Figure 15 shows the effect of different algorithms in processing the QR code. Due to a beam of light passing through in a diagonal direction, the QR code had an increased uneven illumination phenomenon.
- (3) Figure 16 shows the effect of different algorithms in processing the QR code. Two strong lights focused on the QR code made it have different illumination acquisition in different areas.
- (4) Figure 17 shows the effect of different algorithms in processing the QR code. What caused the uneven illumination phenomenon was that a local highlight stayed in the middle part of the QR code. In this case, we can hardly see the modules in the image.
- (5) Figure 18 shows the effect of different algorithms in processing the QR code. The original image was shot under the environment where a glare gathered in the lower right corner of the image. As a result, it is hard to recognize the position detection patterns with our eyes.

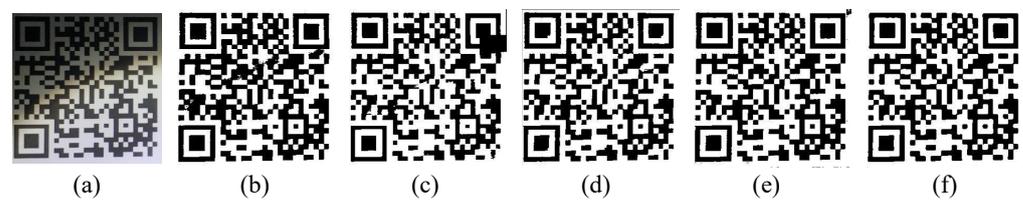


Figure 14. The original image and experimental results of different algorithms: (a) Original image; (b) Sauvola's algorithm [6]; (c) Yao's algorithm [18]; (d) Di's algorithm [8]; (e) Chen's algorithm [13]; (f) Proposed method.

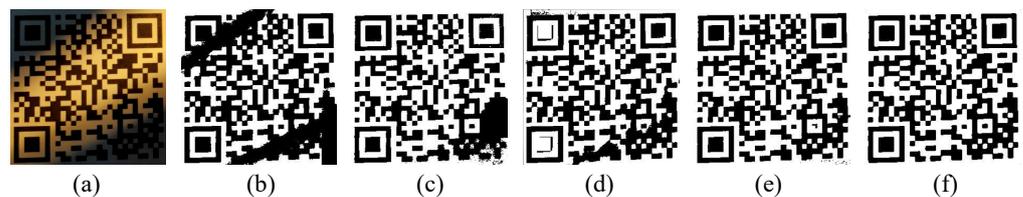


Figure 15. The original image and experimental results of different algorithms: (a) Original image; (b) Sauvola's algorithm [6]; (c) Yao's algorithm [18]; (d) Di's algorithm [8]; (e) Chen's algorithm [13]; (f) Proposed method.

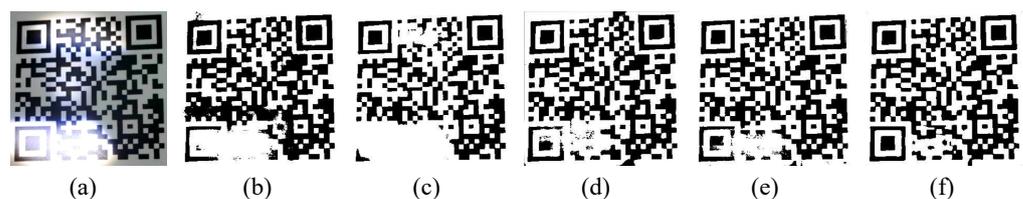


Figure 16. The original image and experimental results of different algorithms: (a) Original image; (b) Sauvola's algorithm [6]; (c) Yao's algorithm [18]; (d) Di's algorithm [8]; (e) Chen's algorithm [13]; (f) Proposed method.

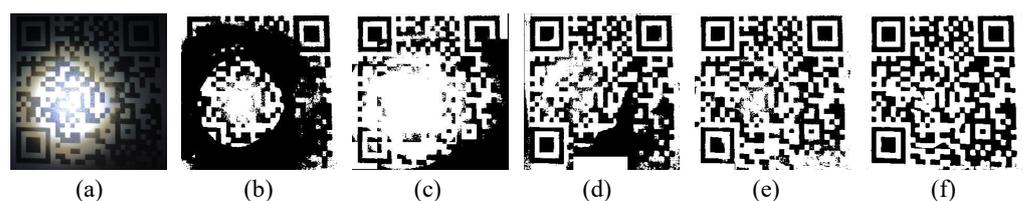


Figure 17. The original image and experimental results of different algorithms: (a) Original image; (b) Sauvola's algorithm; (c) Yao's algorithm; (d) Di's algorithm; (e) Chen's algorithm; (f) Proposed method.



Figure 18. The original image and experimental results of different algorithms: (a) Original image; (b) Sauvola’s algorithm [6]; (c) Yao’s algorithm [18]; (d) Di’s algorithm [8]; (e) Chen’s algorithm [13]; (f) Proposed method.

According to the groups of experiments above, we obtained the values of PSNR and SSIM of different QR codes after using different algorithms. The exact data values are shown in Table 4. In order to compare the values more intuitively, we drew Figure 19 based on the data in Table 4.

Table 4. The values of PSNR and SSIM of different QR codes after using different algorithms.

Sample	Standard	Sauvola	Yao	Di	Chen	Proposed Method
(1)	PSNR	6.4850	6.3371	6.3560	6.5013	6.6319
	SSIM	0.1583	0.1611	0.1505	0.1540	0.1619
(2)	PSNR	7.4802	8.5616	8.0232	8.8290	8.8727
	SSIM	0.3377	0.3590	0.3415	0.3625	0.3649
(3)	PSNR	6.7898	6.4156	7.1323	7.2810	7.4410
	SSIM	0.2698	0.2650	0.2941	0.2999	0.3042
(4)	PSNR	4.0721	4.2207	4.6880	5.5986	5.6867
	SSIM	0.1101	0.1167	0.1072	0.1368	0.1371
(5)	PSNR	4.0956	4.2298	5.0369	4.5181	5.3828
	SSIM	0.1061	0.1058	0.1245	0.1028	0.1404

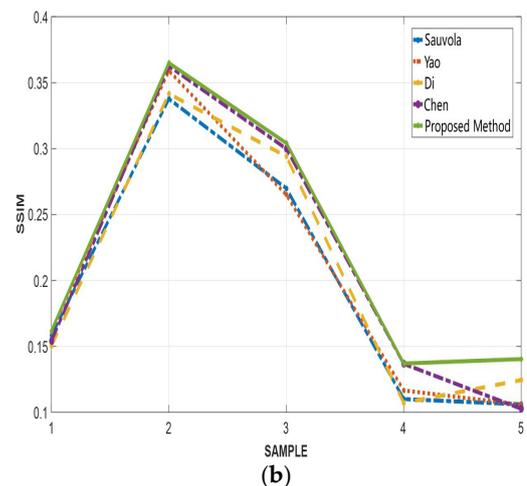
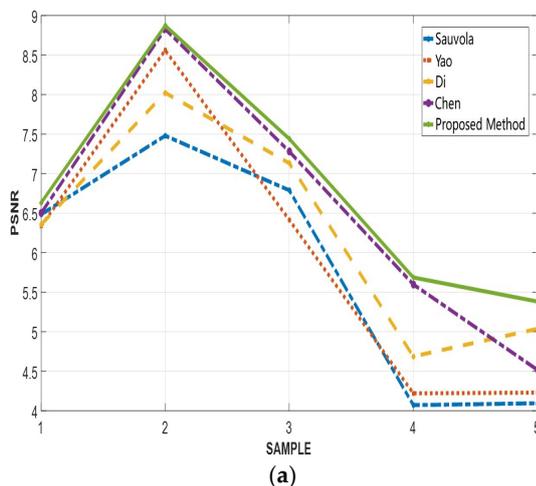


Figure 19. Curves of PSNR and SSIM values of different QR codes after using different algorithms: (a) PSNR; (b) SSIM.

From Figure 14 to Figure 18 above, we can see that the algorithm in this paper is significantly better than several other comparison algorithms. The algorithms of Sauvola and Yao can restore most of the image details in scenes with uneven lighting caused by dim light. However, unrelated dark modules are often introduced in more complex lighting scenes. Furthermore, the QR code processed by the algorithms of Sauvola and Yao has lower image quality. Di’s algorithm has a better processing effect on the types mentioned above of QR codes, but occasionally details will be lost at the part that the light focuses

on. The algorithm that we proposed previously can effectively deal with various types of QR codes, but in the case of identifying position detection patterns difficultly in the image, there will be a small number of details lost. It is worth noting that the PSNR and SSIM values of sample QR codes after using the proposed method in this paper are the largest, according to Table 4 and Figure 17. This proposed method gains an advantage over other comparison algorithms in the image quality of the QR code after binarization. Moreover, the proposed method hardly caused black blocks or information missing, even in the complex lighting environment.

4.2. Recognition Rate and Processing Speed

In order to test the recognition rate and processing speed of various algorithms, the experiment used 30 samples with slightly uneven illumination and 50 samples with strong uneven illumination. The size of the samples is 300×300 . The results of the five algorithms are shown in Table 5.

Table 5. Recognition rates after processed by different algorithms.

Algorithm	Average Time Consume/s	Recognition Rate	
		ZXing	WeChat
None	—	35%	48.75%
Sauvola [6]	4.3871	37.5%	37.5%
Yao [18]	0.0113	37.5%	42.5%
Di [8]	1.5310	53.75%	71.25%
Chen [13]	0.1715	88.75%	92.5%
Proposed Method	0.0611	92.5%	97.5%

It is shown in Table 5 that the recognition rate of Sauvola's algorithms and Yao's are shallow, and the recognition rate of Sauvola's is even less ideal than when no algorithm is applied. The algorithm in this paper uses the integral image algorithm to speed up the threshold value calculation, and the processing speed has been greatly improved, making the processing efficiency much better than Sauvola's. Surprisingly, it is faster than the algorithm we proposed before. Compared with other methods, the algorithm proposed in this paper has the highest recognition rate on ZXing V3.4.0 and WeChat Version 7.0.10.

In a word, the algorithm proposed in this paper has advantages in terms of processing quality, speed, and recognition rate.

5. Conclusions

A new adaptive binarization method was put forward in this paper. This method, combined with the features of the QR code, looked for the position detection patterns to solve the problem of window size definition and dynamically define the size. In addition, using an integral image algorithm in the method significantly improved the speed of threshold calculation. After a lot of experiments and comparison with some excellent current algorithms, the algorithm in this paper has advantages in processing speed and quality of QR codes with uneven illumination. It meets the need for automatic sorting in logistics systems and can also be applied there.

Author Contributions: Conceptualization, R.C., J.X. and X.L.; Funding acquisition, R.C., W.L. and X.L.; Methodology, R.C. and W.L.; Project administration, J.X.; Resources, X.L.; Software, R.C. and W.L.; Validation, W.L. and K.L.; Writing—original draft, W.L. and R.C.; Writing—review and editing, R.C., K.L., J.X., L.W. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62072122, in part by the Special Projects in Key Fields of Ordinary Universities of Guangdong Province under Grant 2021ZDZX1087, in part by the Special Project Enterprise Scitech Commissioner of Guangdong Province under Grant GDKTP2021033100, and the Undergraduate Universities Teaching Quality and Teaching Reform Project Construction Project of Guangdong Province (Yuejiao Gaohan [2021] No. 29)—Science, Industry and Education Integration Practice Teaching Base Construction Project (No. 31).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to appreciate the contributions from Digital Content Processing and Security Technology of Guangzhou Key Laboratory of School of Computer Science, Guangdong Polytechnic Normal University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Centenaro, M.; Costa, C.E.; Granelli, F.; Sacchi, C.; Vangelista, L. A Survey on Technologies, Standards and Open Challenges in Satellite Iot. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1693–1720. [[CrossRef](#)]
2. Laghari, A.A.; Wu, K.; Laghari, R.A.; Ali, M.; Khan, A.A. A Review and State of Art of Internet of Things (IoT). *Arch. Comput. Methods Eng.* **2022**, *29*, 1395–1413. [[CrossRef](#)]
3. Kumar, D.; Singh, R.K.; Mishra, R.; Wamba, S.F. Applications of the Internet of Things for Optimizing Warehousing and Logistics Operations: A Systematic Literature Review and Future Research Directions. *Comput. Ind. Eng.* **2022**, *171*, 108455. [[CrossRef](#)]
4. Su, J.-P.; Wang, C.-A.; Mo, Y.-C.; Zeng, Y.-X.; Chang, W.-J.; Chen, L.-B.; Lee, D.-H.; Chuang, C.-H. I-Logistics: An Intelligent Logistics System Based on Internet of Things. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 331–334.
5. Wellner, P.D. Adaptive Thresholding for the DigitalDesk. *Xerox* **2021**, *EPC-1993-110*, 1–19.
6. Sauvola, J.; Pietikainen, M. Adaptive Document Image Binarization. *Pattern Recognit.* **2000**, *33*, 225–236. [[CrossRef](#)]
7. Zhang, W.; Yang, T. An Improved Algorithm for QR Code Image Binarization. In Proceedings of the 2014 International Conference on Virtual Reality and Visualization, Shenyang, China, 30–31 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 154–159.
8. Di, Y.-J.; Shi, J.-P.; Mao, G.-Y. A QR Code Identification Technology in Package Auto-Sorting System. *Mod. Phys. Lett. B* **2017**, *31*, 1740035. [[CrossRef](#)]
9. Yang, L.; Feng, Q. The Improvement of Bernsen Binarization Algorithm for QR Code Image. In Proceedings of the 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, 23–25 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 931–934.
10. Bernsen, J. Dynamic thresholding of gray-level images. In Proceedings of the Eighth International Conference on Pattern Recognition, Paris, France, 28–31 October 1986.
11. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [[CrossRef](#)]
12. Wu, Y.; Yu, S.; Yang, M. Quick Response Code Binary Research Based on Basic Image Processing. *Sens. Mater.* **2019**, *31*, 859–871. [[CrossRef](#)]
13. Chen, R.; Yu, Y.; Xu, X.; Wang, L.; Zhao, H.; Tan, H.-Z. Adaptive Binarization of QR Code Images for Fast Automatic Sorting in Warehouse Systems. *Sensors* **2019**, *19*, 5466. [[CrossRef](#)] [[PubMed](#)]
14. Ren, J.; Sun, H.; Zhao, H.; Gao, H.; Maclellan, C.; Zhao, S.; Luo, X. Effective Extraction of Ventricles and Myocardium Objects from Cardiac Magnetic Resonance Images with a Multi-Task Learning U-Net. *Pattern Recognit. Lett.* **2022**, *155*, 165–170. [[CrossRef](#)]
15. Sun, G.; Fu, H.; Ren, J.; Zhang, A.; Zabalza, J.; Jia, X.; Zhao, H. SpaSSA: Superpixelwise Adaptive SSA for Unsupervised Spatial-Spectral Feature Extraction in Hyperspectral Image. *IEEE Trans. Cybern.* **2021**, *7*, 6158–6169. [[CrossRef](#)] [[PubMed](#)]
16. Zhou, J.; Liu, Y.; Li, P. Research on Binarization of QR Code Image. In Proceedings of the 2010 International Conference on Multimedia Technology, Ningbo, China, 29–31 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–4.
17. Zhang, Y.; Gao, T.; Li, D.; Lin, H. An Improved Binarization Algorithm of QR Code Image. In Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, 21–23 April 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2376–2379.
18. Yao, S.; Li, P.; He, L.; Li, Y. Uneven Illumination Two-Dimensional Code Image Recognition Algorithm Research. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2043–2046.
19. Mukhopadhyay, S.; Chanda, B. A Multiscale Morphological Approach to Local Contrast Enhancement. *Signal Process.* **2000**, *80*, 685–696. [[CrossRef](#)]
20. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*; Pearson Education India: Delhi, India, 2009; ISBN 81-317-2695-9.

21. Bradley, D.; Roth, G. Adaptive Thresholding Using the Integral Image. *J. Graph. Tools* **2007**, *12*, 13–21. [[CrossRef](#)]
22. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.