

Adaptive Median Filtering Algorithm Based on Divide and Conquer and Its Application in CAPTCHA Recognition

Wentao Ma¹, Jiaohua Qin^{1,*}, Xuyu Xiang¹, Yun Tan¹, Yuanjing Luo¹ and Neal N. Xiong²

Abstract: As the first barrier to protect cyberspace, the CAPTCHA has made significant contributions to maintaining Internet security and preventing malicious attacks. By researching the CAPTCHA, we can find its vulnerability and improve the security of CAPTCHA. Recently, many studies have shown that improving the image preprocessing effect of the CAPTCHA, which can achieve a better recognition rate by the state-of-the-art machine learning algorithms. There are many kinds of noise and distortion in the CAPTCHA images of this experiment. We propose an adaptive median filtering algorithm based on divide and conquer in this paper. Firstly, the filtering window data quickly sorted by the data correlation, which can greatly improve the filtering efficiency. Secondly, the size of the filtering window is adaptively adjusted according to the noise density. As demonstrated in the experimental results, the proposed scheme can achieve superior performance compared with the conventional median filter. The algorithm can not only effectively detect the noise and remove it, but also has a good effect in preservation details. Therefore, this algorithm can be one of the most strong tools for various CAPTCHA image recognition and related applications.

Keywords: Image preprocessing, machine learning, CAPTCHA recognition, adaptive median filtering algorithm.

1 Introduction

The development of CAPTCHA recognition technology not only facilitates the user to obtain massive information, but also promotes the development of technologies such as image processing and pattern recognition. Image denoising is an important part of the CAPTCHA recognition technology. Thus, this paper analyzes the image denoising algorithm with the image CAPTCHA as the research object.

Image denoising algorithm has formed a complete algorithm technology system along with the generation and development of image processing technology [Singh and Shree (2016)]. Its emergence means filtering has driven the development of the linear denoising [Gonzalez and Woods (2002)]. However, such algorithms blur the details of the image

¹ College of Computer Science and Information Technology, Central South University of Forestry & Technology, Changsha, 410114, China.

² Department of Mathematics and Computer Science, Northeastern State University, Oklahoma, 74464, USA.

* Corresponding Author: Jiaohua Qin. Email: qinjiaohua@163.com.

while removing noise, so they were quickly replaced by nonlinear denoising. The nonlinear filtering denoising algorithm [Varade, Dhotre and Paturkar (2013)] can effectively suppress the interference pulse and random noise, which can also preserve the edge information of the image. In recent years, with the development of computer vision technology such as autonomous driving, the real-time requirements for image preprocessing have been constantly improved, but the conventional median filtering has a slow sorting and cannot meet the real-time requirements, so improvements of the median filtering are always the focus of research.

The conventional median filtering used bubble method for sorting pixel values, which needs to sort all pixels in the neighborhood to obtain the median value [Tukey (1974); Pitas and Venetsanopoulos (1992)]. For $N \times N$ filter window, $N^2(N^2 - 1)/2$ comparison operations are required. Taking the 3×3 filter window as an example, the comparison of median value is 36 times, which is a time-consuming process. Brodland et al. [Brodland and Veldhuis (1998)] proposed a weighted median filter, the weight value of the central pixel is defined by the degree of noise pollution. This algorithm can effectively suppress noise and greatly reduce the complexity, which can meet the requirements of computer vision detection system in protecting the edge and details. Huang et al. [Huang, Yang and Tang (2003)] make full use of the data correlation. By considering the relationship between the move-in value, move-out value and median value, which greatly improve the efficiency of the filtering process. On the basis of comparing various fast median filtering, Dai et al. [Dai, Xu and Piao et al. (2017)] proposed an improved fast median filtering algorithm, which combined the sorting algorithm with the hardware system to effectively improve the processing speed. However, these algorithms only do a lot of work in improving the filtering efficiency, but they have not optimized the adaptiveness of filtering window. The window size of the conventional median filtering is fixed, and it is impossible to simultaneously denoise and preserve the image details [Zhang, Xu and Dong (2006)]. Therefore, it is necessary to dynamically alter the size of the window during the filtering process. Zhang et al. [Zhang, Tang and Shi (2014)] proposed a Recursive of Least Square (RLS) adaptive filtering which has good denoising performance and high precision. Bhadouria et al. [Bhadouria, Ghoshal and Siddiqi (2014); Roy, Singha and Manam et al. (2017)] did a lot of optimization in the adaptiveness of filtering window, and also achieved good results, but they have neglected the complexity. Ding et al. [Ding, Niu and Lu et al. (2018); Fan, Han and Gou et al. (2018); Roy, Singha and Devi (2016)] used the convolutional neural network (CNN) and support vector machine (SVM) in the field of image recognition. Although this method has achieved good results, they have not tried to preprocess the original image input by the network, and outstanding image preprocessing algorithm combines with state-of-the-art machine learning algorithm will certainly improve the CAPTCHA recognition accuracy.

In this paper, we propose an adaptive median filtering algorithm based on divide and conquer:

- (1) The conventional median filtering is mostly based on bubble sorting. This paper presents another idea: On the one hand, the median filtering based on the quick sorting algorithm is developed by using divide and conquer, which effectively

reduces the sorting. On the other hand, the data correlation is fully utilized by the methods of literature, which can greatly improve the efficiency of filtering.

- (2) The window size of the conventional median filter is fixed, so it is impossible to simultaneously denoise and preserve image details. Thus, we propose an algorithm that can adaptively adjust the size of the window according to the density of noises, which can not only remove noise but also protect details and avoids refinement or coarsening of edges.

This paper combines the advantages of the above two algorithms and proposes an adaptive median filtering algorithm based on divide and conquer, which can improve the adaptiveness of filtering window and reduce the sorting complexity. The experimental results of peak signal-to-noise ratio (PSNR) and recognition rate show that compared with the conventional median filtering algorithm, the proposed method has a better performance in denoising and image information preservation.

2 Related work

2.1 Preprocessing

In order to facilitate the next denoising, the image needs to be grayed, etc. The RGB image is generally converted into a one-dimensional gray values according to certain coefficients.

$$Gray = \frac{R + G + B}{3} \quad (1)$$

where R , G , and B denote the pixel points in the three-color channels, $Gray$ is the gray value corresponding to the pixel points. The specific calculation method: obtaining the RGB value, taking the average as the gray value of the pixel corresponding to the new image. The grayscale image is G .

Since the background and the character color of CAPTCHA image are quite different, the binarization processing is required as follows.

$$F(x, y) = \begin{cases} 0, & G(x, y) \leq T_0 \\ 255, & G(x, y) > T_0 \end{cases} \quad (2)$$

where $G(x, y)$ denotes the grayscale value at the image (x, y) , and $F(x, y)$ denotes the binarized value. The steps to select the appropriate threshold T_0 are as follows.

Step 1. Set an initial threshold $T_0 = 127$.

Step 2. Each pixel P is divided into character data G_1 and background data G_2 according to the threshold.

Step 3. The average value of G_1 is M_1 , and the average value of G_2 is M_2 .

Step 4. The new threshold is $T_1 = \frac{M_1 + M_2}{2}$.

Step 5. If $T_1 \neq T_0$, turn to Step 2; if equal, T_1 is the final threshold.

After the five steps, the suitable threshold T_0 can be found, and the binarized image is I .

2.2 Closed operation

Morphological processing is a form of domain operation. Through a special definition of a domain called “structural element”, which performs specific logical operation on the region corresponding to the binarized image I at each pixel position. The result of the logical operation is output as the pixel corresponding to the image. The closed operation is used after the image binarization to smooth the outline of the character, suppress the small noise of the character boundary, and fill the small holes between characters.

Structural elements act as the input image to produce the output image in closed operation. The binarized image I and the structural element S used to process I are usually relatively small windows. The closed operation is a process of eroding and corrosion, which is expressed as follows.

$$I \bullet S = (I \oplus S) \ominus S \quad (3)$$

Where I denotes the image to be processed, S denotes the defined structural element.

The functions of closed operation are followed.

- (1) Closed operations can fill small holes, connect adjacent objects, smooth their boundaries, and maintain the position and shape of the object.
- (2) It filters image by filling the concave corners of the image.
- (3) The effect of filtering depends on the size of structural elements.

After that, the image is clearer and the edges smoother. The closed operation is recorded as H .

2.3 Image quality evaluation standard

Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are commonly used to measure image quality.

- (1) Mean Square Error (MSE)

$$MSE = \frac{1}{K \times L} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} (p'(i, j) - p(i, j))^2 \quad (4)$$

Where $p'(i, j)$ and $p(i, j)$ denote the image to be evaluated and the original image, respectively, and K, L denote the length and width of an image.

- (2) Peak Signal-to-Noise Ratio (PSNR)

$$PSNR = 10 \times \lg \frac{Q^2}{MSE} \text{ (dB)} \quad (5)$$

Where Q denotes the gray level after quantization, which is generally 255.

3 Conventional median filter

In the sequence of two-dimensional, a digital image with m rows and n columns can be denoted by $\{f(t, s); t = 1, 2, \dots, m; s = 1, 2, \dots, n\}$. Then the output of the two-dimensional array median filter is

$$O(x, y) = \text{med}\{f(t, s)\}, (t, s) \in A(x, y) \quad (6)$$

where (x, y) denotes a position coordinate, $A(x, y)$ denotes a neighborhood of (x, y) , which contains the size $r \times r$ of the corresponding filter window and $r \times r$ pixels. According to the median filter and the image denoising, the filter can be implemented by the following steps.

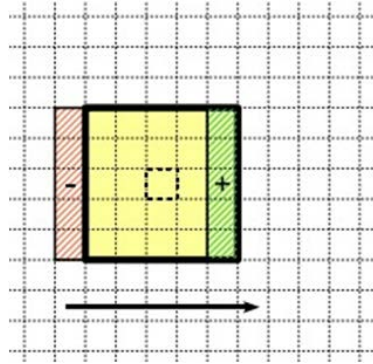


Figure 1: The sliding rule of the median filter window

Step 1. The filtering window traverses the entire image as shown in Fig. 1 and overlaps the corresponding position with its center.

Step 2. Reads each pixel value in the filter window and sorts them.

Step 3. Finds the median value of the array and assign it to the corresponding center filter window pixel.

Algorithm 1: Conventional Median Filtering Algorithm

Input: image H of size $m \times n$, kernel radius r

Output: image O of the same size as H

```

1: for  $i = r$  to  $m - r$  do
2:   for  $j = r$  to  $n - r$  do
3:     initialize array  $A[]$ 
4:     for  $a = i - r$  to  $i + r$  do
5:       for  $b = j - r$  to  $j + r$  do
6:         add  $H(a, b)$  to  $A[]$ 
7:       end for
8:     end for
9:     sort  $A[]$ , then  $O(i, j) \leftarrow \text{med}(A[])$ 
10:  end for
11: end for

```

Conventional median filter has been widely used to eliminate impulse noise which works well at low noise. However, it works not well at high noise density, and the image details such as some threads and sharp corners are destroyed easily in such case that leads to image distortion. Furthermore, it uses the predefined fixed filter window, and pixels are filtered by the windows with the same size. So, it is hard to get better filtering performance when the image has been seriously polluted.

4 Adaptive median filtering based on divide and conquer

The median filtering algorithm has been developed over decades and has achieved encouraging results in denoising, but there are still many disadvantages in processing complex image: 1) The denoising performance is affected by the noise density, and its performance drops sharply when the noise density is large. 2) The details of the images such as thin lines and sharp corners may be erroneously eliminated. 3) The filter window is not flexible enough to automatically adjust the size of the processed image. This paper takes the CAPTCHA images as experimental data to analyze the adaptive median filtering algorithm from two aspects: the filter window adaptability and the improved denoising effect.

4.1 Quicksort by data correlation

Blocks sort is the essence of quicksort based on divide and conquer, which can greatly reduce the complexity of sorting and improve the efficiency of filtering compared with the conventional sorting algorithm. The method quickly sorts the data of the first filter window. The algorithm is described as follows.

Algorithm 2: Quicksort Algorithm Based on Divide and Conquer

Input: unordered array
 output: ordered array
 1: filter window array $A[]$, initialize $\text{left} = 0$, $\text{right} = \text{len}(A) - 1$
 2: $i \leftarrow \text{left}$ && $j \leftarrow \text{right}$
 3: $\text{key} \leftarrow A[\text{left}]$, take the first array element as standard data
 4: while $i < j$
 5: if $i < j$ && $\text{key} \leq A[j]$
 6: $j -= 1$, looking forward from right
 7: $A[i] \leftrightarrow A[j]$
 8: if $i < j$ && $\text{key} > A[i]$
 9: $i += 1$, looking from left to back.
 10: $A[j] \leftrightarrow A[i]$
 11: $A[i] \leftarrow \text{key}$
 12: end while
 13: sort $(A, \text{left}, i - 1)$, sort the small sequences on the left
 14: sort $(A, i + 1, \text{right})$, sort the small sequences on the right
 15: return \rightarrow ordered array $A[]$.

The filter window moves by one pixel next to one pixel, a column of elements are removed and inserted in each side of the window, but most of them remain same. Therefore, we can only consider the influence of the inserted and deleted elements on the median value of the previous window, so that we don't need to compare all the elements, which greatly reduces the sorting complexity.

Assumption that the filter window moves, the column elements inserted on the right side are R_c , $c = 1, 2, \dots, 2n + 1$, the column elements shifted out on the left side are L_c , $c = 1, 2, \dots, 2n + 1$. The algorithm steps are followed.

 Algorithm 3: Sliding Window Pixel Sorting Algorithm Based on Data Correlation

 Input: R_c, L_c , window array $A_c[]$

 Output: the median ($A_c[]$)

 1: Initialize $R_c, L_c, A_c[], c = 1$

2: repeat

 3: Judging whether R_c is equal to L_c .

 4: Calculate $R_c = L_c$, the array $A_c[]$ order is unchanged.

 5: Calculate $R_c \neq L_c$, replacing L_c with R_c . Reordering new array by algorithm 2.

 6: $c = c + 1$

 7: Filter window continues to move and until the entire image is traversed.

4.2 Adaptiveness of filtering window

The performance of the median filter is greatly affected by the size of the filter window. There is a contradiction between noise elimination and detail protection: the smaller filter window can protect some details of images, but the filtering effect on noise should be enhanced. Conversely, the larger window has good noise filtering performances, but it will cause some blurring of images. In addition, according to the median filter principle, if the number of noise points in the filtering window is larger than the number of pixels, the performance of conventional median filter drops dramatically.

In the filtering process, the adaptive median filter will change the size of window according to the preset conditions, and it can also judge whether the current pixel is noise. If the current pixel is noise, then the pixel is replaced by the median value. If not, its current value is maintained. The adaptive median filter has three functions. 1) Filtering salt and pepper noise, 2) smoothing other non-impulse noise, 3) protecting the details of the image as much as possible to avoid the refinement or coarsening of the edges.

The relevant symbols are defined as follows by describing the adaptive median filter algorithm based on divide and conquer. The ordered array $A[]$ in the filter window $S_{r \times r}$ can be obtained quickly by fast sorting method based on divide and conquer, Z_{min} is the minimum value in the filter window, Z_{max} is the maximum value in the filter window, Z_{med} is the median in the filter window, Z_{xy} is the value in (x, y) , S_{max} is the maximum size allowed by filter window $S_{r \times r}$. The formula is followed.

$$A_1 = Z_{med} - Z_{min} \quad (7)$$

$$A_2 = Z_{med} - Z_{max} \quad (8)$$

Formula (7) and (8) determine whether the median value of $S_{r \times r}$ is noise, $A_1 > 0$ and $A_2 < 0$.

$$B_1 = Z_{xy} - Z_{min} \quad (9)$$

$$B_2 = Z_{xy} - Z_{max} \quad (10)$$

If $A_1 > 0$ and $A_2 < 0$ are not satisfied, the median value Z_{med} of $S_{r \times r}$ is noise. In this case, formula (9) and (10) are used to gradually adjust the filter window size to find the suitable non-noise median.

 Algorithm 4: Adaptive Median Filtering Based on Divide and Conquer

Input: image H of size $m \times n$, kernel radius r Output: image O of the same size as H

```

1: for  $i = r$  to  $m - r$  do
2:   for  $j = r$  to  $n - r$  do
3:     initialize array  $A[]$ 
4:     for  $a = i - r$  to  $i + r$  do
5:       for  $b = j - r$  to  $j + r$  do
6:         add  $H(a, b)$  to  $A[]$ 
7:         get the ordered array  $A[]$  quickly by algorithm2,3, where
            $Z_{min} \leftarrow \min(A[])$ ,  $Z_{max} \leftarrow \max(A[])$ ,  $Z_{med} \leftarrow \text{med}(A[])$ 
8:       end for
9:     end for
10:     $A_1 \& A_2$  are given by formula (7)(8) and  $B_1 \& B_2$  are given by formula
       (9)(10)
11:    if  $A_1 > 0 \ \&\& \ A_2 < 0$ ,  $Z_{med}$  is noise or not.
12:      if  $B_1 > 0 \ \&\& \ B_2 < 0$ , whether  $Z_{xy}$  is noise.
13:        return  $O(i, j) \leftarrow Z_{xy}$ , output  $(x, y)$  directly
14:      else
15:        return  $O(i, j) \leftarrow Z_{med}$ 
16:      end if
17:    else
18:       $S_{r \times r} += 2$ ,  $Z_{med}$  is noise, extend the size of  $S_{r \times r}$ 
19:      if  $S_{r \times r} \leq S_{max}$ ,  $S_{r \times r}$  is less than the maximum size of  $S_{max}$ 
20:        extend the filter window size, repeat 4~16.
21:      else
22:        return  $O(i, j) \leftarrow Z_{med}$ 
23:      end if
24:    end if
25:  end for
26: end for

```

According to the algorithm, the adaptive median filter algorithm based on divide and conquer has great innovation, which not only greatly simplifies the sorting but also makes the size of filtering window more flexible. Especially in the flexibility, the system can change the size of the filter window automatically with the noisy density, which greatly reduces the human intervention, thereby improving the performance.

5 Experimental results and analysis

In order to evaluate the performance of the proposed algorithm, we use four kinds of the CAPTCHA image experimental data provided by Inspur Group for denoising and compare it with the conventional median filtering. The experiment was carried out on Windows 64, CPU 8 GHz, 4 G RAM computer. Firstly, describe the CAPTCHA image to

be used in this experiment briefly. There are 4 types of 10,000 CAPTCHA images for each class.

- (1) Four arithmetic expressions (including numbers and operators, resolution 350×80);
- (2) Lowercase English letters and numbers (resolution 200×60);
- (3) Slightly overlapped uppercase English letters and numbers (resolution 200×80);
- (4) The Chinese character string with complicated geometric structure and uneven stroke size (resolution 150×45).

5.1 Results of related preprocessing

Image preprocessing involves multiple steps, and this paper focuses on the filter denoising module. Three related preprocessing effects are shown in Fig. 2.

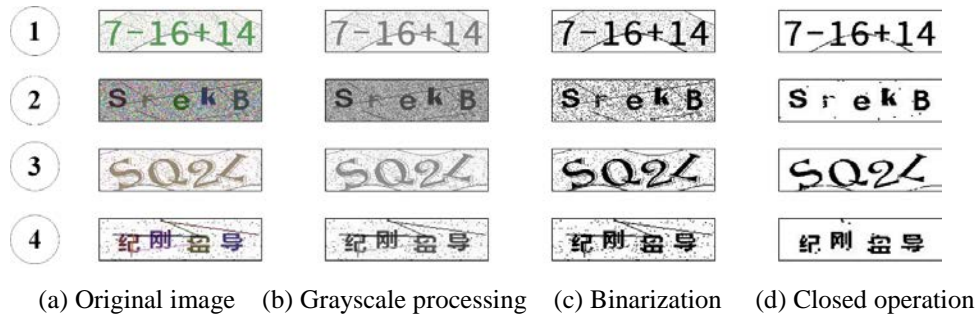


Figure 2: Comparison of the effects of related work

In Fig. 2, after three preprocessing operations (b), (c), and (d), the noise of CAPTCHA images has been removed a lot, especially after the closed operation. Although the contour is clear, there are still problems such as line noise and stroke blur in the detail of images, which need to be filtered.

The results of the two filtering algorithms to CAPTCHA images are shown in Fig. 3.

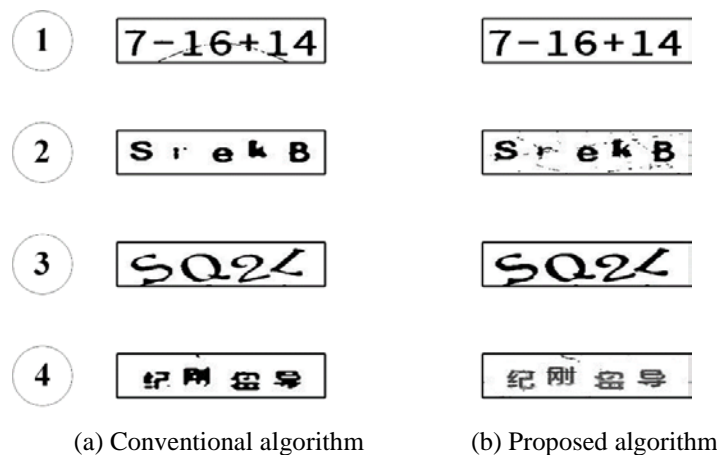


Figure 3: Comparison of two filtering denoising algorithms

From Figs. 3(a) and 3(b), we can see that the denoising performance of the proposed algorithm is better than the conventional median filtering algorithm. Dealing with the first category of CAPTCHA images, the string is non-tilt, non-adhesive and the overall contour is clear. The performances of the two algorithms are very good, and the adaptive median filtering algorithm is completely denoised.

For the second category, the outline of characters is clear and regular. The algorithm has not only good denoising performance but also preserves fine stroke information. Although the denoising effect of the conventional algorithm is better, it is obviously not desirable to destroy the image information.

For the third category, although the overall outline of characters is clear the characters are slightly stuck and tilted, the overall performance of the two algorithms is very good, but the proposed algorithm performs better in smoothing images.

For the fourth category, the geometric structure of Chinese characters is complicated and the thickness of strokes is uneven, which leads to a slightly blurred image of the relevant preprocessing. Under a bad condition, the algorithm of this paper can still present good performances. Compared with the conventional algorithm, this algorithm destroys the image information less, which can preserve the main information of the image to the utmost extent.

5.2 Performance Comparison

Considering improving the recognition rate of CAPTCHA, there are two evaluation criteria, one is the peak signal-to-noise ratio, and the other is the recognition rate of the CAPTCHA image by the tiny convolutional neural network.

Table 1: Comparison of PSNR by two filtering algorithms

CAPTCHA category	Conventional algorithm (dB)	Proposed algorithm (dB)
First category	32.53	32.71
Second category	33.27	32.33
Third category	31.88	31.11
Fourth category	32.71	51.63

From Tab. 1, the quality of the adaptive median filtering algorithm based on divide and conquer is better than the conventional median filtering, especially the fourth category.

Table 2: Comparison of PSNR by proposed algorithm and closed operation

CAPTCHA category	Closed operation (dB)	Proposed algorithm (dB)
First category	32.49	32.71
Second category	33.86	32.33
Third category	31.07	31.11
Fourth category	32.82	51.63

From Fig. 2, the outline of CAPTCHA image after the closed operation is very clear. The experimental results in Tab. 2 present that the adaptive filtering is still necessary, which can improve the images quality.

CAPTCHA image denoising aims to improve the recognition rate, and the recognition used the state-of-the-art machine learning algorithm-convolutional neural network. According to the difference of CAPTCHA image, training network models can suit their respective characteristics. Then, it can identify different filtered images to obtain the recognition rate.

The first three types of CAPTCHA image only include numbers, operators, and letters. Therefore, 7000 images from these three categories were randomly selected for preprocessing, and then the convolutional neural network model was trained. The remaining 3000 images are processed by two different filtering methods and inputted into the trained model to compare the recognition rate.

The fourth type of CAPTCHA image only includes Chinese characters, and its network model structure must be different from the first three types. After preprocessing, 7000 images are randomly selected and directly input to the network model for training. Since the rotated Chinese characters are particular in the feature tensor generated by the whole image, the model can be used to correctly recognize Chinese characters. Then, the remaining 3000 images are processed by two different filtering methods and then input them into the trained model to compare the recognition rate.

Table 3: Comparison of recognition rates by various algorithms

CAPTCHA category	Conventional algorithm (%)	Proposed algorithm (%)
First category	99.37	100
Second category	83.47	96.53
Third category	71.47	96.81
Fourth category	91.37	100

From Tab. 3, the image processed by the adaptive median filter algorithm based on divide and conquer has a significant improvement in recognition rate. In the experiment, there are clear strokes about the first and fourth types of CAPTCHA, so the effects of two algorithms on recognition rate are basically similar. However, for the second and third types of CAPTCHA with blurred strokes and handwriting, the performance of adaptive median filter algorithm based on divide and conquer is much better than the conventional median filter about recognition rate. Although this algorithm performs very well in the processing of CAPTCHA images of the second and third categories, the recognition rate still can't reach 100%, which is one of the next research directions

6 Conclusions

This paper proposes an improved median filter algorithm based on divide and conquer. On the one hand, the filter window can automatically adjust size by the density of noise to achieve good denoising performance and preserve the information of the images to the utmost extent. On the other hand, the algorithm can quickly calculate the median value of

the first filter window by using the divide and conquer method without sorting all the data, which can make full use of the correlation of data, calculate the median value of the remaining windows in turn, and greatly improve the filtering efficiency. Although the algorithm has achieved good results in the experiment, the denoising effect is poor for images with blurred writing and oblique strokes, which leads to bad recognition performance. For the future work, we will further improve the filtering performance under higher noise intensity and severe distortion.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (No. 61772561), the Key Research & Development Plan of Hunan Province (No. 2018NK2012), the Postgraduate Research and Innovation Project of Hunan Province (No. CX2018B447), the Postgraduate Science and Technology Innovation Foundation of Central South University of Forestry and Technology (20183027), the Key Laboratory for Digital Dongting Lake Basin of Hunan Province.

References

- Bhadouria, V. S.; Ghoshal, D.; Siddiqi, A. H.** (2014): A new approach for high density saturated impulse noise removal using decision-based coupled window median filter. *Signal Image Video Process*, vol. 8, no. 1, pp. 71-84.
- Brodland, G. W.; Veldhuis, J. H.** (1998): Three-dimensional reconstruction of live embryos using robotic microscope images. *IEEE Transactions on Biomedical Engineering*, vol. 45, no 9, pp. 1173-1181.
- Dai, S. B.; Xu, W.; Piao, Y. J.; Chen, Y. T.** (2017): Design and implementation of fast median filtering algorithm based on system generator. *Journal of Computer Applications*, vol. 34, no. 8, pp. 2518-2520.
- Ding, M. Y.; Niu, Y. L.; Lu, Z. W.; Wen, J. R.** (2018): A method for identifying commodity parameters in images based on deep learning. *Journal of Software*, vol. 29, no. 4, pp. 1039-1048.
- Fan, W.; Han, J. G.; Gou, F.; Li, S.** (2018): Recognition of Chinese character CAPTCHA by convolutional neural network. *Computer Engineering and Applications*, vol. 54, no. 3, pp. 160-165.
- Gonzalez, R. C.; Woods, R. E.** (2002): *Digital Image Processing*. Prentice Hall, New Jersey, USA.
- Huang, T.; Yang, G.; Tang, G.** (2003): A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13-18.
- Li, H.; Qin, J. H.; Xiang, X. Y.; Ma, W. T.; Xiong, N. N.** (2018): An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE ACCESS*, vol. 6, pp. 66963-6697.
- Pitas, I.; Venetsanopoulos, A. N.** (1992): Order statistics in digital image processing. *Proceedings of the IEEE*, vol. 80, no 12, pp. 1893-1921.

Ravikumar, G.; Dr, V. S. (2017): A machine learning approach for MRI brain tumor classification. *Computers, Materials & Continua*, vol. 53, no. 2, pp. 91-108.

Roy, A.; Singha, J.; Devi, S. S. (2016): Impulse noise removal using SVM classification based fuzzy filter from gray scale images. *Signal Processing*, vol. 128, no. C, pp. 262-273.

Roy, A.; Singha, J.; Manam, L.; Laskar, R. (2017): Combination of adaptive vector median filter and weighted mean filter for removal of high-density impulse noise from color images. *IET Image Processing*, vol. 11, no. 6, pp. 352-361.

Singh, K. M. (2012): Vector median filter based on non-causal linear prediction for detection of impulse noise from images. *International Journal of Computational Science and Engineering*, vol. 7, no. 4, pp. 345-355.

Singh, P.; Shree, R. A. (2016): Comparative study to noise models and image restoration techniques. *International Journal of Computer Applications*, vol. 149, no. 1, pp. 18-27.

Tukey, J. W. (1974): Nonlinear (nonsuperposable) methods for smoothing data. *Proceedings of the IEEE Electronics and Aerospace Systems Conference*, vol. 673, no. 1, pp. 673-681.

Varade, M. R. R.; Dhotre, M. R.; Pahurkar, M. A. B. (2013): A survey on various median filtering techniques for removal of impulse noise from digital images. *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 2, pp. 606-609.

Xie, B.; Qin, J. H.; Xiang, X. Y.; Li, H.; Pan, L. L. (2018): An image retrieval algorithm based on GIST and SIFT features. *International Journal of Network Security*, vol. 20, no. 4, pp. 609-616.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2018): Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*. <https://doi.org/10.1007/s11063-018-9892-7>.

Xiang, L. Y.; Zhao, G. H.; Li, Q. (2018): TUMK-ELM: a fast unsupervised heterogeneous data learning approach. *IEEE Access*, vol. 6, pp. 35305-35315.

Zhang, R.; Tang, Q. W.; Shi, C. M. (2014): Image denoising based on recursive least squares adaptive filtering algorithm. *Journal of Computer Applications and Software*, vol. 31, no. 4, pp. 193-196.

Zhang, X. M.; Xu, B. S.; Dong S. Y. (2006): Adaptive switch median filtering for removing impulsive noise. *Optoelectronic Engineering*, vol. 33, no. 6, pp. 78-83.