VOL. 20, NO. 3, 2021, 15-25
www.elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Efficient Thinning Algorithm for Malaysian Car Plate Character Recognition

Saifullahi Sadi Shitu[1*], Syed A.R. Abu-Bakar[2], Nura Musa Tahir[3], U. I. Bature[4], Haliru Liman[5]

[1]Cyber Security Department, Nigerian Defence Academy
[2]School of Electrical Engineering, Faculty of Engineering, UTM
[3]Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University, Bauchi, Nigeria
[4]Department of Computer and Communication Engineering, Abubakar Tafawa Balewa University, Bauchi, Nigeria
[5]Department of Electrical and Electronics Engineering, Abubakar Tafawa Balewa University, Bauchi, Nigeria

[*]Corresponding author: saifullahisadi22@gmail.com

Abstract: The thinning algorithm is one of the approaches of identifying each character printed on the car plate. Malaysian car plate characters appear in different character sizes, styles, customized printed characters etc. These variations contribute to difficulty in thinning successfully segmented and extracted license plate characters for recognition. To address these problems, an improved thinning operation for Malaysian car plate character recognition is proposed. In this algorithm, samples from segmented and extracted license plates are used for a thinning operation which is passed to Zhang-Suen thinning algorithm that could not guarantee one pixel thick and then to single pixelate algorithm that provides one pixel width of character for recognition. From the simulation, the result obtained has clearly proven to be the best for character recognition systems with least number of white pixels (777 pixels) and 0.26% redundant pixel left in the medial curve.

Keywords: Thinning algorithm, Malaysian car plate, Car plate character, License plate, Pixel, Single pixelate.

Article History: received 29 August 2021; accepted 27 October 2021; published 27 December 2021.

1. INTRODUCTION

License Plate Recognition (LPR) system is a system that utilizes image processing techniques and character recognition algorithms to identify vehicles by localizing their license plates and then recognize each character printed on the plate [1, 2]. The rapid growth in human population and economic development have led to the increase in the number of vehicles annually [3]. Therefore, LPR has an important role in many applications such as in parking management systems, traffic surveillance system, toll gate and highway checking, speed checking, vehicle location and navigation, security and public safety [1-4].

LPR is generally divided into three stages after image acquisition, i.e. localization stage, character segmentation stage, and character recognition stage [1-3]. The characters printed on the license plate are usually extracted from the first two stages, and served as the input to the recognition stage. It is well known that the prime issues in the recognition stage are the effectiveness and efficiency of extracting the distinctive feature from the patterns. Most of the recognition algorithms for recognizing alphanumeric characters, binary and ideographs like Stroke analysis [5, 6] requires a thinned character with one pixel width. However, obtaining character with one-pixel thickness and noise free is itself a challenge. Therefore, efficient thinning, or also called skeletonizing algorithm, is needed for obtaining characters with one-pixel thickness [6]. Although the character is one pixel thick, it should maintain all its properties such as position, size and orientation of the original character. Thinning is thus one of the most critical pre-processing steps in any recognition analysis applications. The success of recognition depends directly on how well the thinning algorithm yields accurate outcomes [6-8].

Several extraction techniques have been proposed to realise character recognition process. Almustafa proposed a position of peak algorithm in [9], the system can detect English numbers but it cannot differentiate Arabic numbers of 6, 1 and 0. Another method is the Hierarchical Temporal Memory (HTM) spatial pooler proposed in [10]. Even though the method is accurate and efficient, it has some limitations in detecting some characters. Character recognition based on neural network method was proposed by Wijetunge and Ratnaweera in [11]. They resized each character into a 10×14 pixels and then shaved into a pixel. In their approach, each character was trained with at least ten samples. However, a simple feature used is not enough to differentiate between similar characters.

It is observed that the formation of the license plate characters differs from one country to another. Malaysian car plate characters, for example, do not, in general, conform to the base standard as the characters appear in different sizes, styles, and sometimes with customized printed characters. These variations contribute to difficulty in the thinning process and consequently, hinders a successful character recognition. Moreover, some of these characters such as “D”, ”0”, “Q”, “W”, “M”, “5”, “S”, “X”, “K”, “8”, and “B” have some similarities that cause difficulty to recognise after the thinning process. The significant of any thinning algorithm is to represent a character with contiguous one pixel width, that preserves the shape of the character, topology of the original character, and robust against noise [12].

The approach undertaken in this paper is inspired by the recognition process developed by Tan [13], but with some modification in the thinning process. The thinning process in [13] was based on Zhang and Suen [14] algorithm. One major drawback of Zhang-Suen (ZS) algorithm is in its failure to guarantee one-pixel width of the character. However, the ZS algorithm provides better shape and maintains the connectivity of the thinned character. Hence, we utilized the algorithm proposed by Guo and Hall [15] as a correction stage to improve the ZS thinning algorithm. This algorithm guarantees one-pixel width of the character. However, this algorithm alone compromises the character shape, even though it is good for single pixel width character.

This work aimed at improving the thinning algorithm for recognizing normal Malaysian car plate characters using any recognition algorithm. The work is based on offline images extracted from the video streams recorded at Universiti Teknologi Malaysia (UTM) main entrance. The organization of this paper is as follows: Section 1 provides the introduction along with some previous related works. Section 2 provides the methodology used in realising the proposed research. Section 3 illustrates the result and discussion, and finally, Section 4 presents the conclusion and recommendation of the work.

2. METHODOLOGY

This section discusses the entire process flow to ensure successful recognition of Malaysian license plate characters. Figure 1 shows the general recognition overview of this work.

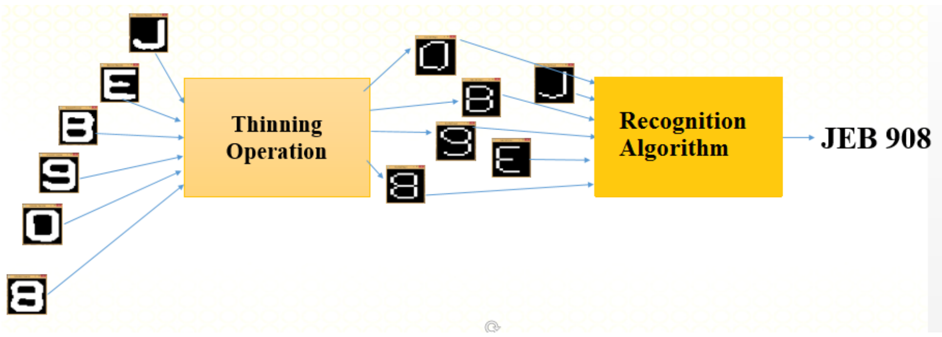


Figure 1. General overview of the recognition system

The algorithm used by [15] is adopted as a correction stage to improve the ZS thinning algorithm. The input to this process is the extracted binary character obtained from the preceding stages. Prior to the thinning operation, the extracted binary character is first resized into a template dimension of 11×17 pixel before being passed to the recognition stage. Figure 2 illustrates the entire process flow of the recognition stage along with the one of the recognition algorithms block diagram.

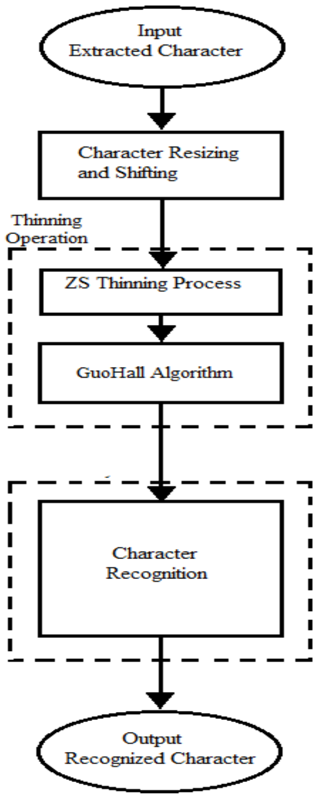


Figure 2. Thinning and recognition process

2.1 Character Resizing and Shifting

The template dimension used for matching the extracted character in this work is 11×17 pixels. However, the dimension of the input extracted characters usually varies and is not fixed. Therefore, to avoid tracing error (trace out of the template) during character tracing resizing the character is necessary. However, the width of some of the characters is smaller than the template size which may tamper the properties of the original character such as “I” or “1” after thinning. Therefore, the extracted character is first copied and then pasted to the centre of a blank image in order to avoid this problem.

Besides resizing, character shifting also plays an important role to avoid error in tracing the character. In the case of extracted character, the character may be out of position from the center of the image after thinning operation. Retaining the location of the thinned character may cause the algorithm to fail in recognizing the actual character. Therefore, the character needs to be shifted accordingly to the center of the image for character tracing. In short, character resizing and shifting are a two-way process. The character might need to be resized and then shifted to the center for error avoidance to enhance the entire recognition performance.

2.2 Character Thinning Operation

Character thinning is a fundamental pre-processing step in many character recognition algorithms which consists of simplifying foreground pixel (white pixel) until the character is one pixel thick (skeleton). As mentioned earlier, the algorithm for thinning operation is based on ZS algorithm.

2.2.1 Zhang Suen (ZS) Thinning Operation

This consists of removing all the contour points of the character except those points that belong to the skeleton. In this algorithm two sub-iterations are used, one is targeted to remove the south-east boundary points and north-west corner points while the other is targeted to remove the north-west boundary points and south-east corner points. Figure 3 shows the pixel kernel used to fulfil the operation.

p_8 $(i-1, j-1)$	p_1 $(i-1, j)$	p_2 $(i-1, j+1)$
p_7 $(i, j-1)$	p (i, j)	p_3 $(i, j+1)$
p_6 $(i+1, j-1)$	p_5 $(i+1, j)$	p_4 $(i+1, j+1)$

Figure 3. Pixel kernel design

Supposed the known target (object) marked as 1 (white pixel), and background point as 0 (black pixel). Based on the conditions given below, the contour point located at p_1 is deleted from the original image during the first iteration.

1. $2 \le B(p) \le 6$
2. $C(p) = 1$
3. $p_1 \times p_3 \times p_5 = 0$
4. $p_3 \times p_5 \times p_7 = 0$
5. $p_1 \times p_3 \times p_7 = 0$
6. $p_1 \times p_5 \times p_7 = 0$

where $B(p)$ is the number of non-zero (white pixel) adjacent neighbors of p which is defined in condition 1, and $C(p)$ is the number of occurrence of 0 to1 patterns in the sequential manner of all the neighbors of p from p_1 to p_8. If none of the condition is satisfied, then contour point p remains. The third and fourth conditions changed in the second sub iteration to condition five and six respectively.

The purpose of condition (1) is to preserve the endpoints of a skeleton line and that of condition (2) is to prevent the point lies in between the endpoints of a skeleton line being removed. There is a possibility that the removed point might be located at the south-east boundary or north-west corner point if p_3 or p_5 equals to zero or p_1 and p_7 equal to zero. Similarly, point p that is removed in the second sub iteration using condition (5) and (6) might be located at north-west boundary point or south-east corner point.

2.2.2 Single pixelated character

Since the above approach cannot guarantee one pixel thickness for the thinned character, the result of ZS thinning process is further passed to the GuoHall's algorithm 1 (A1) [17] to remove all the redundant pixels. There are two sub-iterations in this algorithm:

i) Odd iteration which aims at identifying and removing pixels on the north and east boundary of the character except those pixels that belong to the skeleton.
ii) Even iteration which aims at identifying and removing pixels on the south and west boundary of the character except those pixels that belong to the skeleton.

This algorithm is also based on a binary image containing of 1s and 0s with the foreground (character to be deleted) pixels being 1 (white pixel) while the background pixels being 0 (black pixel). For implementing the operation, similar kernel as shown in Figure 3 is used in this algorithm. The symbols □, ∪ and ∩ used in this algorithm denote logical complement, OR and AND respectively, and reserve + for arithmetic addition.

The target point p is deleted from the character if the following conditions are satisfied;

For odd iteration
(a) $C(p) = 1$
(b) $2 \le N(p) \le 6$
(c) $(p_1 \cup p_2 \cup \sim p_4) \cup p_3 = 0$

For even iteration
(a) $C(p) = 1$
(b) $2 \le N(p) \le 3$
(c) $(p_5 \cup p_6 \cup \sim p_8) \cap p_7 = 0$
where $N(p) = \min[N_1(p), N_2(p)]$

$$N_1(p) = (p_8 \cup p_1) + (p_2 \cup p_3) + (p_4 \cup p_5) + (p_6 \cup p_7)$$
$$N_2(p) = (p_1 \cup p_2) + (p_3 \cup p_4) + (p_5 \cup p_6) + (p_7 \cup p_8)$$
$$C(p) = \sim p_1 \cap (p_2 \cup p_3) + \sim p_3 \cap (p_4 \cup p_5) + \sim p_5 \cap (p_6 \cup p_7) + \sim p_7 \cap (p_8 \cup p_1)$$

Thinning operation is repeated until there is no further redundant pixel to be deleted. Condition (a) preserves local connectivity when p is deleted and avoids deletion of pixels in the middle of the medial curves, it also allows some of the ones in the middle of the double pixels diagonal lines to be deleted. In (b) the variable $N(p)$ allows endpoints to be preserved while deleting many redundant pixels in the middle of the curves. Therefore, this algorithm solves the problem associated with ZS algorithm and produces thinner result without any redundant pixel.

3. RESULTS AND ANALYSIS

This section presents results and analysis of the combination of ZS, Guo-Hall and single pixelate algorithms with respect to the extracted characters from

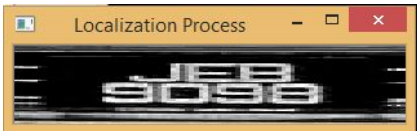
Malaysian license plate numbers. In this work, the algorithms were implemented in C++ language running on Windows 8.1 platform, with 4.00 GB RAM and Intel core i5 processor with 2.20 GHz CPU.

3.1 Sample Data

The extracted characters used in this work are obtained from ninety-two (92) images extracted from the video stream of vehicles entering and leaving the UTM main entrance. Some samples of localized plate number images used in this paper are depicted in Figure 4. Table 1 represents the segmented and extracted characters from the plate number images. These characters are used to evaluate each thinning algorithm used in this work. The best-thinned character will be used for recognition analysis technique.

Figure 4. Samples of localized plate numbers

Table 1: The segmented and extracted characters from localized plate number images.

s/n	Localized plate number	Extracted characters
1		
2		
3		
4		

3.2 Evaluation Performance Measure

Both the ZS and Guo-Hall algorithms are compared with Single Pixelate Algorithm (SPA). The relative algorithm performance is compared in two ways;

1. Observation was made considering whether the character looks like the original character.
2. After simulating the algorithm over several different test sets the following measure were carried out:

i) $R1$, defined as the size of the medial curve in number of white pixel (255) as illustrated in Figure 9.

ii) $R2$, defined as the size of redundant pixels in the medial curve, the rectangular pixel in Figure 9 is a redundant pixel.

iii) $R3$, defined as the percentage of redundant pixels left in the medial curve, that is

$$R3 = \frac{number\ of\ redundant\ pixels\ in\ the\ medial\ curve}{total\ number\ of\ pixels\ in\ the\ medial\ curve} \times 100\%$$

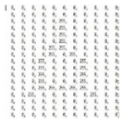
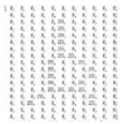
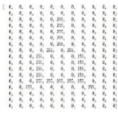
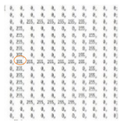
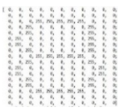
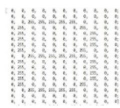
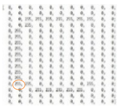
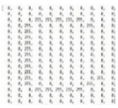
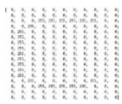
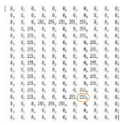
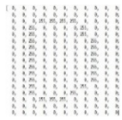
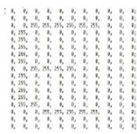
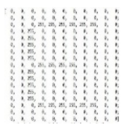
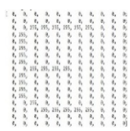
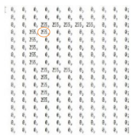
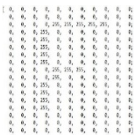
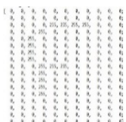
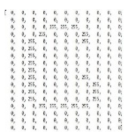
Where a redundant pixel is defined as a pixel in the medial curve, which is not an end point, the deletion of which did not disconnect the curve and the number of redundant pixels is defined as the maximum number of redundant pixels that can be removed simultaneously without disconnecting the medial curve. Figure 5 shows examples of redundant pixels.

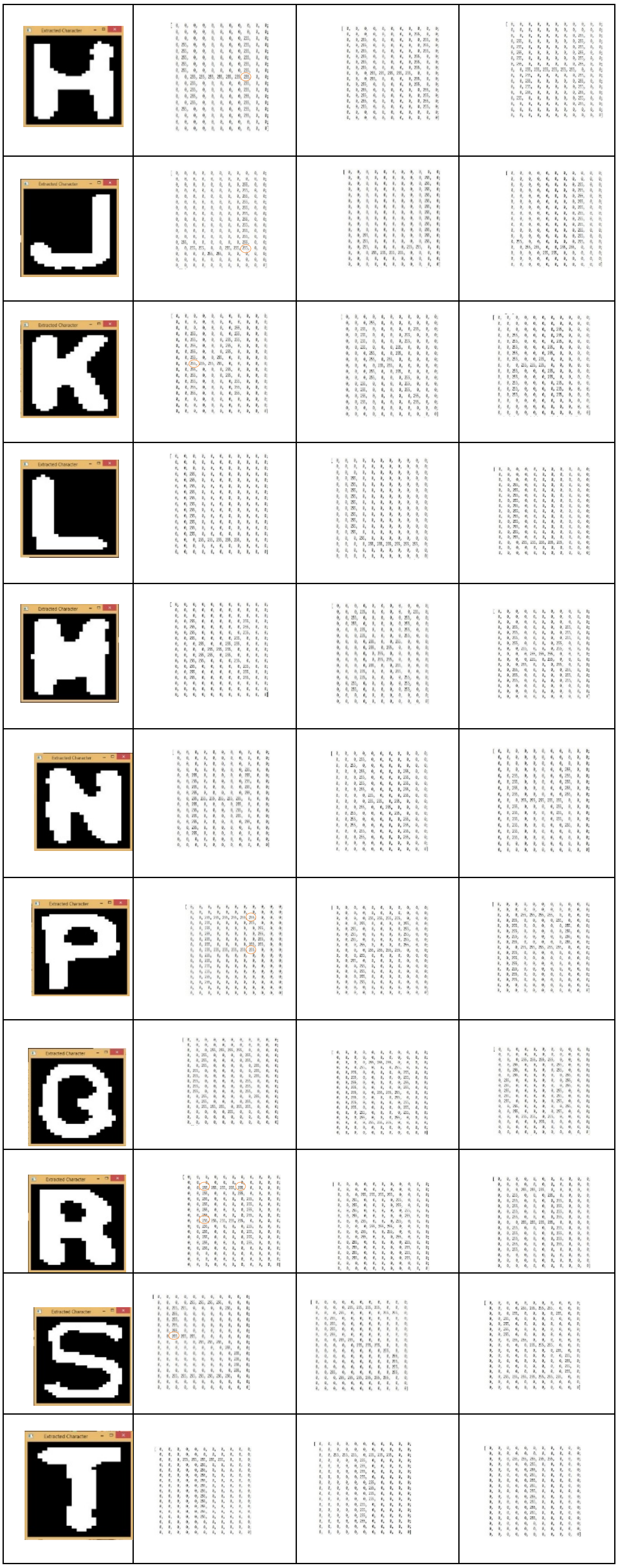
The first test of characters and corresponding medial curves of each algorithm are given in Table 2. Pixels of the character (foreground) are denoted by "255" and the deleted pixel/background is denoted with "0".

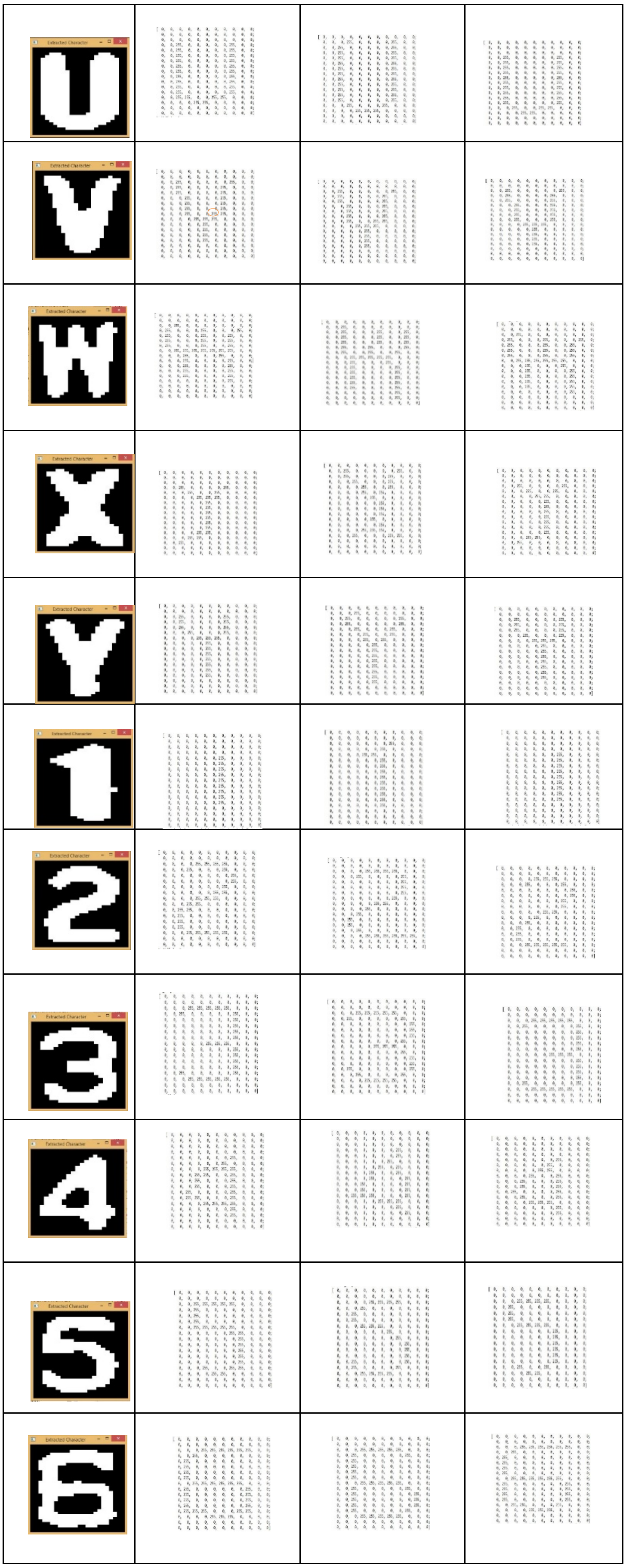
```
[ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;
  0,   0, 255, 255, 255, 255, 255, 255,   0,   0,   0;
  0, 255,   0,   0,   0,   0,   0, 255,   0,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0, 255,   0,   0,   0;
  0, 255, 255, 255, 255, 255, 255, 255,   0,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0,   0, 255,   0,   0;
  0, 255,   0,   0,   0,   0,   0, 255, 255,   0,   0;
  0,   0, 255, 255, 255, 255, 255,   0,   0,   0,   0;
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0]
```

Figure 5. Pixel in a rectangular box is a redundant pixel

Table 2. Thinning comparison for the three algorithms over various alphanumeric characters extracted from Malaysian car plate numbers based on the shape of the Characters.

Extracted Characters	Algorithm		
	ZS	GuoHall	SPA (Proposed)
			
			
			
			
			
			
			





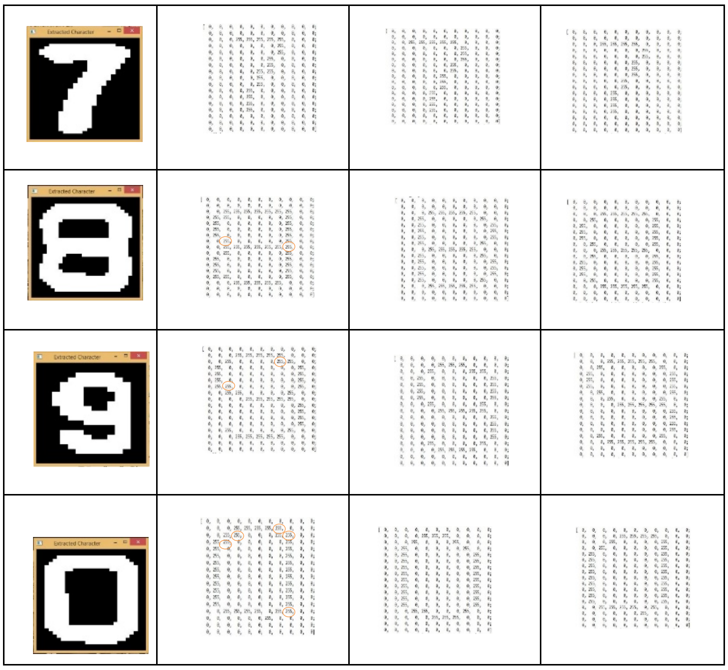


Table 2 presents the thinning results of the ZS, Guo-Hall and SPA algorithms for different alphanumeric characters extracted from Malaysian plate numbers. Based on the results shown in Table 2, ZS thinning result appears to maintain the shape of the original character. However, there are some redundant pixels in which removing these pixels will not disconnect the medial curve, as shown by the circled pixels in ZS column in Table 2. Furthermore, these redundant pixels might confuse the recognition algorithm, and hence, may lead to unsuccessful or erroneous recognition.

On the other hand, Guo-Hall thinning results appeared thinner. However, the shape of some characters are compromised, in the sense that the character looks entirely different from the original character, example letters are “A, B, C, D, K” from Guo-Hall column in Table 2. It is evident from the table that the proposed SPA thinning algorithm shows the best performance such that each character is thinner and does maintain the structure of original character. This makes the result of SPA suitable for recognition systems.

Table 3 shows the thinning performance for the three sets of algorithms. It can be seen that SPA able to successfully thin each character with lowest redundancy index, while none of the two algorithms can achieve this.

Another interesting observation is that SPA produces result with the least or no redundant pixels with the least number of white pixels (255). ZS algorithm produces the most redundant pixels. For instance, in case of character “4” up to 21% of the pixels are redundant. Naturally, the ZS algorithm produces the highest number of white pixel. Conversely, Guo-Hall performance result shows that all the redundant pixels are successfully dealt with. However, the number of white pixel (R1) in some thinned characters is still higher compared to that of SPA thinned characters.

Table 4 summarizes the performance of all the three algorithms based on the total white pixels, total redundant pixels and the percentage redundant pixels left of all the extracted characters used. It is clear that SPA is proven to be the best for character recognition systems than the two algorithms (ZS and Guo-Hall) with least number of white pixels (777 pixels) and 0.26% redundant pixel left in the medial curve. ZS algorithm produces about 8.18% redundant pixels and highest number of white pixels (844). While on the other hand Guo-Hall produces 0.37% redundant pixels and 809 white pixels which are comparably more than that of SPA.

.

Table 3. Thinning performance for three thinning algorithms over alphanumeric characters.

Characters	Algorithms		
	ZS	GuoHall	SPA
A	$R1 = 23$ $R2 = 1$ $R3 \approx 4\%$	$R1 = 23$ $R2 = 0$ $R3 = 0$	$R1 = 22$ $R2 = 0$ $R3 = 0$
B	$R1 = 38$ $R2 = 3$ $R3 \approx 8\%$	$R1 = 35$ $R2 = 0$ $R3 = 0$	$R1 = 35$ $R2 = 0$ $R3 = 0$
C	$R1 = 24$ $R2 = 2$ $R3 \approx 8\%$	$R1 = 24$ $R2 = 0$ $R3 = 0$	$R1 = 22$ $R2 = 0$ $R3 = 0$
D	$R1 = 31$ $R2 = 2$ $R3 \approx 4\%$	$R1 = 28$ $R2 = 0$ $R3 = 0$	$R1 = 29$ $R2 = 0$ $R3 = 0$

E	$R1 = 26$ $R2 = 1$ $R3 \approx 4\%$	$R1 = 26$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$
F	$R1 = 18$ $R2 = 1$ $R3 \approx 6\%$	$R1 = 18$ $R2 = 0$ $R3 = 0$	$R1 = 16$ $R2 = 0$ $R3 = 0$
G	$R1 = 24$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$	$R1 = 24$ $R2 = 0$ $R3 = 0$
H	$R1 = 29$ $R2 = 1$ $R3 \approx 3\%$	$R1 = 30$ $R2 = 0$ $R3 = 0$	$R1 = 28$ $R2 = 0$ $R3 = 0$
J	$R1 = 19$ $R2 = 1$ $R3 \approx 5\%$	$R1 = 21$ $R2 = 0$ $R3 = 0$	$R1 = 18$ $R2 = 0$ $R3 = 0$
K	$R1 = 26$ $R2 = 2$ $R3 \approx 8\%$	$R1 = 27$ $R2 = 0$ $R3 = 0$	$R1 = 24$ $R2 = 0$ $R3 = 0$
L	$R1 = 16$ $R2 = 0$ $R3 = 0$	$R1 = 17$ $R2 = 0$ $R3 = 0$	$R1 = 16$ $R2 = 0$ $R3 = 0$
M	$R1 = 26$ $R2 = 3$ $R3 \approx 12\%$	$R1 = 27$ $R2 = 0$ $R3 = 0$	$R1 = 23$ $R2 = 0$ $R3 = 0$
N	$R1 = 25$ $R2 = 1$ $R3 = 4\%$	$R1 = 27$ $R2 = 0$ $R3 = 0$	$R1 = 24$ $R2 = 0$ $R3 = 0$
P	$R1 = 29$ $R2 = 4$ $R3 \approx 14\%$	$R1 = 24$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$
Q	$R1 = 30$ $R2 = 2$ $R3 \approx 7\%$	$R1 = 30$ $R2 = 0$ $R3 = 0$	$R1 = 28$ $R2 = 0$ $R3 = 0$
R	$R1 = 28$ $R2 = 2$ $R3 \approx 7\%$	$R1 = 29$ $R2 = 0$ $R3 = 0$	$R1 = 26$ $R2 = 0$ $R3 = 0$
S	$R1 = 29$ $R2 = 2$ $R3 \approx 7\%$	$R1 = 28$ $R2 = 0$ $R3 = 0$	$R1 = 27$ $R2 = 0$ $R3 = 0$
T	$R1 = 16$ $R2 = 1$ $R3 \approx 6\%$	$R1 = 18$ $R2 = 0$ $R3 = 0$	$R1 = 16$ $R2 = 0$ $R3 = 0$
U	$R1 = 26$ $R2 = 1$ $R3 \approx 4\%$	$R1 = 29$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$
V	$R1 = 22$ $R2 = 2$ $R3 \approx 9\%$	$R1 = 23$ $R2 = 2$ $R3 \approx 9\%$	$R1 = 21$ $R2 = 1$ $R3 \approx 5\%$
W	$R1 = 31$ $R2 = 2$ $R3 \approx 6\%$	$R1 = 35$ $R2 = 0$ $R3 = 0$	$R1 = 29$ $R2 = 0$ $R3 = 0$
X	$R1 = 18$ $R2 = 2$ $R3 \approx 11\%$	$R1 = 22$ $R2 = 0$ $R3 = 0$	$R1 = 16$ $R2 = 0$ $R3 = 0$
Y	$R1 = 18$ $R2 = 1$ $R3 \approx 6\%$	$R1 = 19$ $R2 = 0$ $R3 = 0$	$R1 = 18$ $R2 = 0$ $R3 \approx 6\%$
1	$R1 = 9$ $R2 = 0$ $R3 = 0$	$R1 = 13$ $R2 = 0$ $R3 = 0$	$R1 = 9$ $R2 = 0$ $R3 = 0$
2	$R1 = 26$ $R2 = 5$ $R3 \approx 19\%$	$R1 = 23$ $R2 = 0$ $R3 = 0$	$R1 = 21$ $R2 = 0$ $R3 = 0$
3	$R1 = 25$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$	$R1 = 25$ $R2 = 0$ $R3 = 0$
4	$R1 = 24$ $R2 = 5$ $R3 \approx 21\%$	$R1 = 21$ $R2 = 1$ $R3 \approx 5\%$	$R1 = 19$ $R2 = 0$ $R3 = 0$

5	$R1 = 25$ $R2 = 5$ $R3 = 20\%$	$R1 = 23$ $R2 = 0$ $R3 = 0$	$R1 = 20$ $R2 = 0$ $R3 = 0$
6	$R1 = 33$ $R2 = 2$ $R3 \approx 6\%$	$R1 = 31$ $R2 = 0$ $R3 = 0$	$R1 = 31$ $R2 = 0$ $R3 = 0$
7	$R1 = 17$ $R2 = 2$ $R3 \approx 12\%$	$R1 = 17$ $R2 = 0$ $R3 = 0$	$R1 = 15$ $R2 = 0$ $R3 = 0$
8	$R1 = 40$ $R2 = 6$ $R3 = 15\%$	$R1 = 35$ $R2 = 0$ $R3 = 0$	$R1 = 34$ $R2 = 0$ $R3 = 0$
9	$R1 = 36$ $R2 = 3$ $R3 \approx 8\%$	$R1 = 31$ $R2 = 0$ $R3 = 0$	$R1 = 33$ $R2 = 0$ $R3 = 0$
0	$R1 = 37$ $R2 = 4$ $R3 \approx 11\%$	$R1 = 31$ $R2 = 0$ $R3 = 0$	$R1 = 33$ $R2 = 0$ $R3 = 0$

Table 4. Performance summary

Characters	Algorithms		
	ZS	GuoHall	SPA
Total $R1$	844	809	777
Total $R2$	69	3	2
Total $R3$	$\frac{69}{844}\times 100\% \approx 8.18\%$	$\frac{3}{809}\times 100\% \approx 0.37\%$	$\frac{2}{777}\times 100\% \approx 0.26\%$

4. CONCLUSION

The fundamental aim of this work is to improve the thinning operation for Malaysian car plate character recognition. Efficient thinning algorithm is an algorithm capable of producing character with one pixel wide, preserved connectivity and look like the original character. It has been shown that combining ZS with GH algorithms produced characters with aforementioned features which are suitable for character recognition systems.

4. ACKNOWLEDGEMENT

We wish to express our deepest appreciation to all those who helped in our study, either directly or indirectly. Among them are, Dr. Rabiu Musa Kwankwaso (former Kano state governor) for financing the whole work and all staff of the School of Electrical Engineering, Universiti Teknologi Malaysia.

REFERENCES

[1] Youting, Zhao, Yu Zhi, and Li Xiying. "Evaluation methodology for license plate recognition systems and experimental results." *IET Intelligent Transport Systems* 12, no. 5, 375-385. 2018.

[2] M. Y. Arafat, A. S. M. Khairuddin and R. Paramesran, "Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework," in *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 712-723, 2020.

[3] H. Tao and A. K. Gopalakrishnam, "License plate extraction and recognition of a Thai vehicle based on MSER and BPNN," in *Knowledge and Smart Technology (KST), 2015 7th International Conference on*, 2015, pp. 48-53. 2015.

[4] Greati, V. R., Ribeiro, V. C. T., da Silva, I. M. D., & de Medeiros Martins, A. A Brazilian license plate recognition method for applications in smart cities. In *2017 IEEE First Summer School on Smart Cities (S3C)* (pp. 43-48). IEEE. 2017.

[5] Gao, Fei, YichaoCai, Yisu Ge, and Shufang Lu. "EDF-LPR: a new encoder–decoder framework for license plate recognition." *IET Intelligent Transport Systems* 14, no. 8, 959-969. 2020.

[6] Bataineh, Bilal. "An iterative thinning algorithm for binary images based on sequential and parallel approaches." *Pattern Recognition and Image Analysis* 28, no. 1, 34-43. 2018.

[7] Saha, P. K., Borgefors, G., &di Baja, G. S. A survey on skeletonization algorithms and their applications. *Pattern recognition letters*, *76*, 3-12. 2016.

[8] Liu, L., Chambers, E. W., Letscher, D., &Ju, T. A simple and robust thinning algorithm on cell complexes. In *Computer Graphics Forum*Oxford, UK: Blackwell Publishing Ltd.Vol. 29, No. 7, pp. 2253-2260, 2010.

[9] Almustafa, Khaled M., Rached N. Zantout, and Hasan R. Obeid. "Peak position recognizing characters in Saudi license plates." In *2011 IEEE GCC Conference and Exhibition (GCC)*, pp. 186-189. IEEE, 2011.

[10] Afeefa, P. P., and Pillai Praveen Thulasidharan. "Automatic License Plate Recognition (ALPR) using HTM cortical learning algorithm." In 2017 International Conference on Intelligent Computing and Control (I2C2), pp. 1-4. IEEE, 2017.

[11] Zhifan, Feng, and Fang Kangling. "Research and implementation of an improved license plate recognition algorithm." In *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 4, pp. 2300-2305. IEEE, 2011.

[12] Chatbri, H., &Kameyama, K. Using scale space filtering to make thinning algorithms robust against noise in sketch images. *Pattern Recognition Letters*, *42*, 1-10. 2014.

[13] Tan, J. L. *Framework for Automatic Malaysian License Plate Identification System* (Master dissertation, Universiti Teknologi Malaysia). 2013.

[14] Zhang, T. Y., & Suen, C. Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, *27*(3), 236-239. 1984.

[15] Guo, Z., & Hall, R. W. (1989). Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, *32*(3), 359-373. 1989.

[16] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM,* vol. 27, pp. 236-239, 1984.

[17] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Communications of the ACM,* vol. 32, pp. 359-373, 1989.