Jurnal
Teknologi

Full Paper

ELECTRONIC ONLINE HANDWRITING CHARACTER RECOGNITION SYSTEM USING ARDUINO PLATFORM

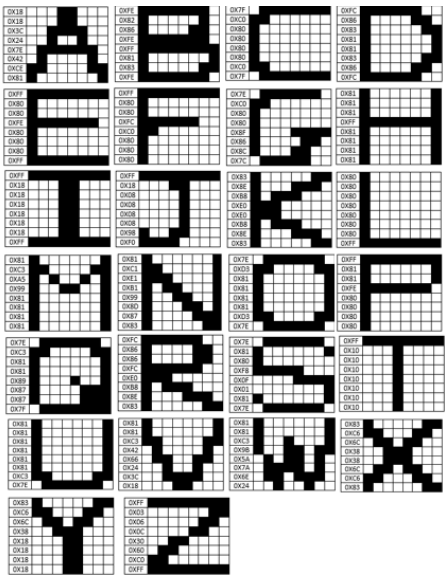
A. R. Syafeeza, Albert Ngan Ban Chew, Muhammad Muhaimin Hilmi Sadari, Mohamad Nasriq Kamaruddin, Wong Jia Li, Zarini Md Zauber

Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), 76100 Durian Tunggal, Melaka, Malaysia

Article history
Received
20 December 2016
Received in revised form
21 June 2017
Accepted
10 September 2017

*Corresponding author
syafeeza@utem.edu.my

Graphical abstract



Abstract

This paper proposes an online prototype approach of handwriting character recognition system using a microcontroller. The main contribution of this work is the simple method of translating a handwriting input from analog to a digital font using XOR bit-wise operation. This system recognizes characters and numbers in natural handwriting with a stylus. This electronic system is used to improve the technology from the use of a bulky keyboard system to a portable and convenient way which is more suitable for smaller electronic devices nowadays such as smart phone, tablet and etc. In order to construct the system, a Arduino Mega is used as the microcontroller along with the Arduino TFT Touch Shield and an LCD screen as the input and output respectively as the hardware components. For the software part, this system uses C language and Arduino software (IDE). The recognition result achieved is 80.0%.

Keywords: Handwriting recognition system, Arduino mega board, Arduino TFT touch screen, XOR bit-wise, character

Abstrak

Makalah ini mencadangkan pendekatan prototaip dalam talian bagi sistem pengenalan aksara tulisan tangan menggunakan mikropengawal. Sumbangan utama kerja ini adalah kaedah mudah untuk menerjemahkan input tulisan tangan dari analog ke fon digital menggunakan operasi XOR secara-bit. Sistem ini mengiktiraf aksara dan nombor dalam tulisan tangan semula jadi dengan stilus. Sistem elektronik ini digunakan untuk meningkatkan teknologi daripada penggunaan sistem papan kekunci yang besar dengan cara mudah alih dan mudah yang lebih sesuai untuk peranti elektronik yang lebih kecil pada masa kini seperti telefon pintar, tablet dan sebagainya. Untuk membina sistem, Arduino Mega digunakan sebagai mikrokontroler bersama-sama dengan Arduino TFT Touch Shield dan skrin LCD sebagai input dan output masing-masing sebagai komponen perkakasan. Bagi bahagian perisian, sistem ini menggunakan perisian bahasa C dan Arduino (IDE). Hasil pengecaman yang dicapai adalah 80.0%.

Kata kunci: Sistem pengecaman tulisan tangan, Arduino mega board, Arduino TFT skrin sentuh, XOR secara-bit, aksara

1.0 INTRODUCTION

Nowadays, people and technologies are literally inseparable. In this modern day, due to the challenging scientific nature of the problem and its industrial importance, many types of research are driven to refine the handwriting recognition system to perfection. The handwriting recognition system is becoming more common as time passes. The handwriting recognition system is the ability of an electronic system to convert handwritten input from paper document as well as touch screen devices to digital font. It is user friendly towards people who do not grow fond of using the keyboard. The handwriting recognition system can also reduce miscommunication in work. It also might offer advantages like ensuring accurate medical care, reduced storage costs, and preserve a vital field of research available for students in the future.

The most challenging part of implementing this system is to improve the accuracy of recognition for various type of handwriting. The handwriting recognition system in this paper is online handwriting recognition system. Online handwriting recognition is a system where by it interfaces specifically a pen or stylus for the user to write on a touch sensitive surface, which may be integrated with an output display. Besides that, a software application is included to interpret the movements of the stylus across the writing surface and translating the resulting strokes into digital text.

In order to develop a handwriting recognition system, it requires implementation of both hardware and software. The hardware part consists of integrating the Arduino Mega board as the microprocessor, Arduino TFT Touch Screen, and LCD screen together whereas the software part uses the Arduino IDE software. The language used for the software in the system will be in C language.

The handwriting recognition system can be done in many ways, either by implementing a different type of algorithms or even different types of sensors to detect the handwriting. Priyanka. M *et al.* [1] did their online handwriting recognition by using an Arduino Uno as the microcontroller and instead of using the conventional touch screen, they used the accelerometer sensor. They used accelerometer-based gesture recognition in which the system uses the gesture path of the hand to recognize the characters. In other words, the accelerometer detects slight movements of the hand in writing and from there on it will predict the character which is written.

K. P. Ambika *et al.* [2], uses the algorithm called bitwise XOR operation in their recognition system. Although their research does not directly relate to handwriting character recognition, the XOR algorithm is easy to implement since it does not require any complex equations or mathematical equations but just a simple XOR. In their research, the XOR algorithm is used to find a matching pattern of certain DNA sequence search in bioinformatics. The algorithm uses simple XOR to find the matching pattern, as XOR logic gives an output '1' when different results are matched whereas output '0' when same results are matched. The work in [3] also used XOR algorithm and the hardware platform is ATmega32.

Rachana R. Herekar and S.R. Dhotre [4], did the offline handwriting character recognition by based on zoning using Euler numbers. The process begins by dividing the handwritten character into 9 zones and classifies each character using Euler number along with using end points features and aspect ratio. The end point features of a character were used to solve the problem arise from different character/alphabet yielding the same Euler number. The aspect ratio was used to solve problem arise from using the end point of a character/alphabet as there is character/alphabet that has the same end points.

Olarik Surinta *et al.* [5], used two types of recognition for recognizing handwritten Bangla digits. The types of recognition were divided into two groups mainly featured based and pixel based. The featured based recognition used was Contour Angular Technique (CAT) and the hotspot technique whereas the pixel based recognition was the Gray Pixel-Based Method and Black and White Down-Scaled. CAT primarily finds the contour of the handwritten image as the feature. The Hotspot Technique is a technique where the distance between evenly spaced hotspots and the closest black pixels in each direction is used to describe the whole handwritten image. The Gray Pixel-Based Methods uses the raw pixel intensities of the handwritten image to preserve the handwritten image whereas the Black and White Scaled Down is the handwritten image being scaled down to a 9×9 resolution. All these 4 methods will then undergo training through Support Vector Classification (SVM). All four techniques produce recognition rate of over 90%.

Farha Mendi *et al.* [6], combines three methods which are Shape Template Matching, Pixel Density Matching, and Stroke Movement Matching. Shape Template Matching matches the shape of the scaled down handwritten character to a preset template, the probability of the correct character recognized would increase if there is a match. Pixel Density Matching uses the variation of density of pixels for each character at a specific position. The density of pixels of the handwritten character is matched to a template, to find the approximate equal. If it matched correctly then the probability for recognizing that specific character will increase. Lastly, Stroke Movement Matching is the type of recognition where it matches the movement of the strokes of the template with the user input. All three results were then combined to yield the result for the recognized character.

J.Pradeep *et al.* [7] uses diagonal based feature extraction in an offline handwriting recognition system. Diagonal base done by them is by dividing the input area of 90×60 pixels size into 54 equal zones where each zone consisting of 10×10 pixels, and features of each zone are extracted by having 19 diagonals lines in the zone, and the foreground pixels

present long each diagonal line is summed to get a single sub-feature. Thus, every zone would have only 1 feature. With a total of 54 zones, it yields 54 features with an addition of 15 features whereby 6 are from horizontally and 9 are from vertically. These 64 features would then undergo training in a neural network for character recognition.

Other existing works have implemented neural network approach for various character recognition applications [8-10] and other type of recognition applications [11-13]. Arduino microcontroller also has been used in other type of recognition applications as reported in [14-15].

The remainder of this paper is organized as follows. In Section 2, we discuss the proposed approaches. In Section 3, we describe the prototype development. This is followed by results and discussion, in Section 4. Finally, we conclude this paper with a summary of the work.

2.0 METHODOLOGY

2.1 Handwriting Recognition

Handwriting Recognition System is an electronic system that can read a handwritten input from papers or photographs as well as touch-screens devices and translates into a digital font. The system is mostly used to translate paper documents into digital documents to save space, environment and also money due to the advantages of changing things into digital form. In addition, it can also help miscommunication in work especially in the medical field.

2.2 Type of Recognition System

There are two types of recognition system which are online recognition and offline recognition.

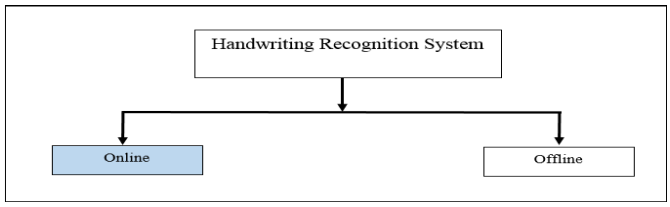


Figure 1 Types of Handwriting Recognition System

Online recognition is the type of recognition where it requires a pen or a stylus to interact with a touch-sensitive surface or a touch screen. Online recognition requires the user to write on it, unlike offline recognition where it requires a ready sample image as the input. Online recognition is a real-time recognition as it translates and recognizes the character directly after writing it. Hence, online recognition will allow the user to convert a handwriting to a digital character immediately. This system can also be found in applications in smart phones as well. This is the type of recognition used in this paper.

Offline recognition is the type of recognition where conversion of text in an image file are converted into letter codes which are usable in another system such as a computer. The input for offline recognition can be regarded as a static representation of handwriting. Offline recognition need not be real-time as it does not require a real-time input such as touch sensors. Input for offline recognition is normal images.

2.3 Method of Handwriting Character Recognition

The type of handwriting character recognition used here is the pixel based method XOR bitwise operation which is highlighted with blue color in Figure 2. As shown in the figure, the method for handwriting character recognition can be classified into two groups namely featured-based and pixel-based. Featured based involves in finding the characteristics of the character and then comparing the characteristics. Pixel based involves comparing of the pixel by pixel of the handwritten input to the template/preset character. XOR bitwise operation is a pixel based method, thus it uses XOR logic to compare pixel by pixel of the handwritten input to the preset a letter or template.

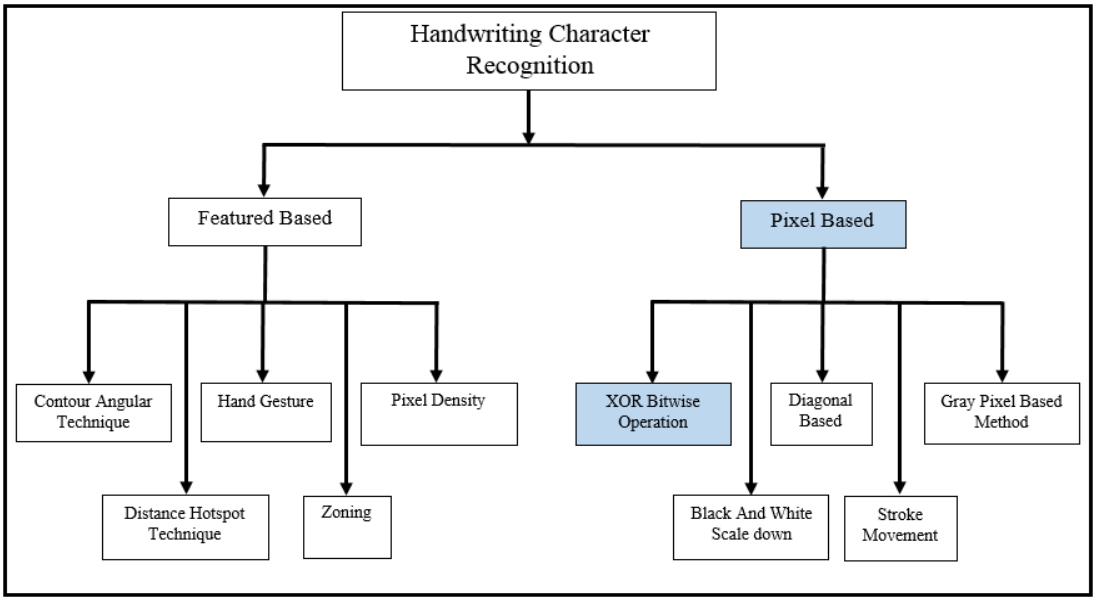


Figure 2 Method of Handwriting Character Recognition

2.4 Prototype Development

In the project, Arduino Mega is used as the microcontroller to process the recognition while the input and output are the Touch Screen and the LCD display respectively. Figure 3 shows the basic block diagram of the system.

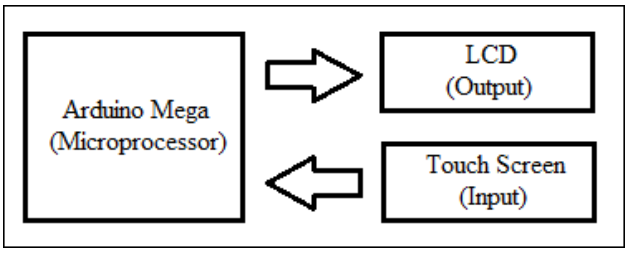


Figure 3 Basic Block Diagram of the System

Based on the block diagram, it can be seen that the touch screen serves as the input whereby every point touched by a finger or a stylus on the screen will be saved in the Arduino Mega. The Arduino Mega will then process the saved points and recognize the pattern through recognition. Thus, the recognized letter or number will then be displayed on the LCD.

2.4.1 Hardware Implementation

This section discusses the circuits and the connections that have been made to establish the communication between the microcontroller and the input/output peripherals. The components involved are Arduino Mega, TFT Touch shield, and LCD. Figure 4 shows the circuit connection of the system and Figure 5 depicts the prototype of the overall system.

The Touch Shield has a resolution of 240×320. A portion of the area of the touch screen are used as touch buttons, hence there is only 240×200 resolution left to be used as the touch screen. The touch screen uses digital pins from pin number 5 to 13 and analog pins from 0 to 3. There is a total of 13 pins but the rest of the unused pins are covered by the shield, therefore, it is hard to use those remaining pins. The touch shield is a 4-wire resistive touch screen.

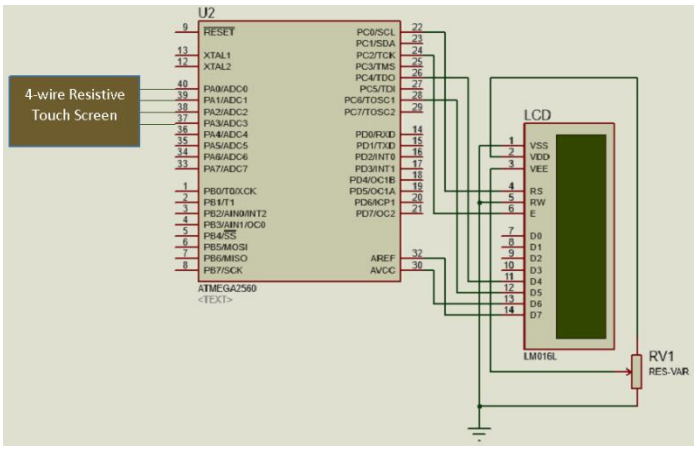


Figure 4 Circuit Connection of the System

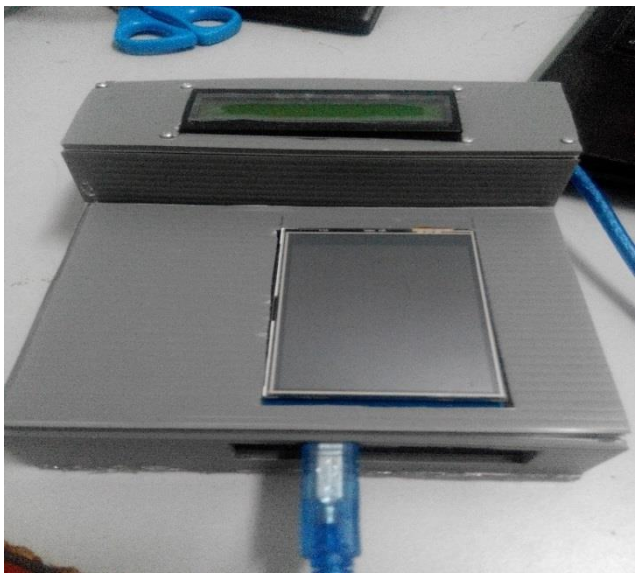


Figure 5 Prototype of the Handwriting Recognition System

According to Figure 4, the LCD screen connected directly to the pins of the microcontroller with the connection. The pin configuration can be referred to Table 1.

Table 1 Configuration of LCD pins

LCD pin	Arduino pin
1	GND
2	5V
3	Middle of Potentiometer
4	22
5	GND
6	24
7	-
8	-
9	-
10	-
11	26
12	28
13	30
14	32
15	Resistor then to 5V
16	GND

All LCD operations are carried out based on the configuration. The words displayed on the LCD is controlled by the software in the microcontroller.

2.4.2 Software Implementation

The software for this system includes the driver or firmware which enables the communication between the hardware devices. However, since Arduino has integrated those drivers or firmware into their own library, therefore the drivers or firmware need not be written by ourselves since it is already provided in the Arduino IDE. The recognition algorithm is the bitwise XOR operation since it is easy to be implemented. Figure 6 shows the flowchart of the whole system.

From the figure, the main code of the program is shown where it will initialize the hardware and save any available input if exist, else it will just keep reading the touch screen for any input. If there is any touched input, the touched point will be saved and the timer interrupt would be reset. The process repeats itself until there is no longer any input touched point being read. The lower picture shows the flowchart of the timer interrupt. The timer interrupt will only start when no handwriting is detected for a period of time where the time is predetermined in the code. Whenever the timer interrupts, it will scale the available input then followed by recognizing the scaled input and lastly display the recognized value on the LCD screen.

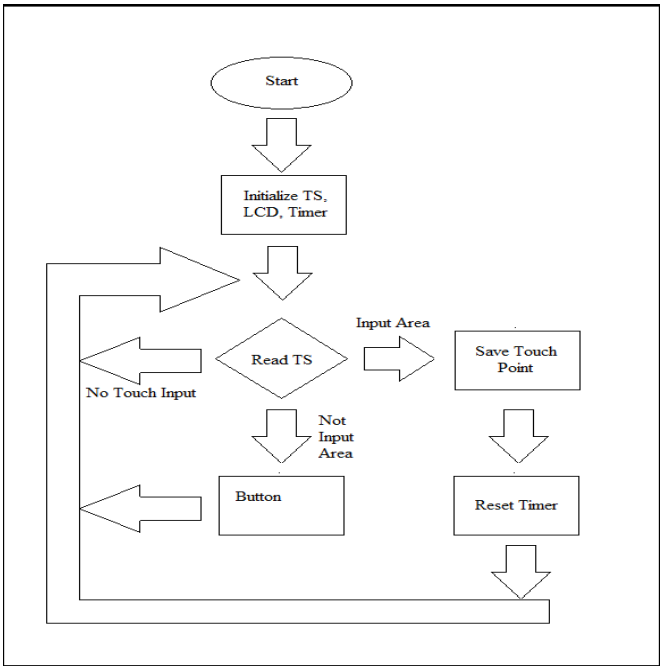


Figure 6 Flowchart of the whole system

Based on the flow chart in Figure 7, the software mainly consist of 3 parts which are input, scaling, and recognition. The input touch screen used is the Arduino 2.8" TFT Touch Shield 2.8. It's a touch screen which is compatible with Arduino Mega and the fact that it's a shield hence no soldering is required except for just mounting the shield on the Arduino Mega. The touch screen does more than only receiving inputs but it also has its own graphics library. The touch screen and graphics library are open sourced therefore the code to use the graphics and reading touch points are done by just simply calling the functions given in the open source library. The touch screen would display a few buttons for the user to the interface by using the graphics library and the touch points would be detected by using the code given in the library. For every point touched on the screen, the points would be saved in an array in the Arduino Mega.

The touch screen would detect the written input until it stops detecting for a short amount of time. After it

stops detecting for a period of time it will start scaling and then proceeds to the recognition of the handwritten input. The Arduino Mega detects the input by using the timer interrupt function to only interrupt the code after it receives no input for a period of time.

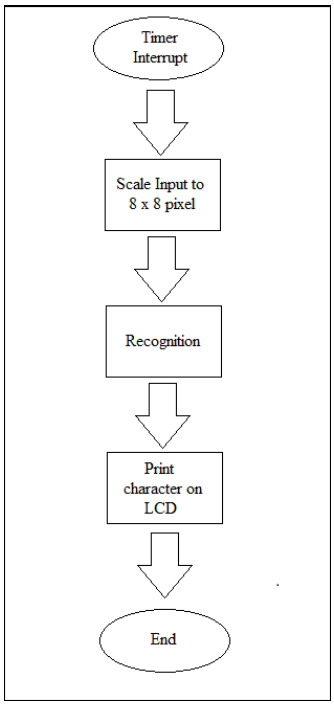


Figure 7 Flowchart of the software system

Figure 8 shows that the written handwriting on the left side which is the resolution of 240×200 is converted to a resolution of 8×8 by the scaling process.

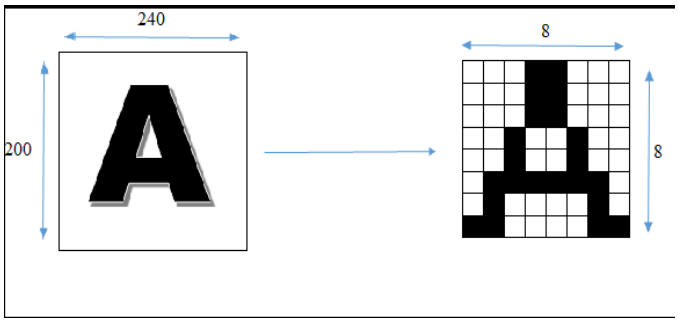


Figure 8 Scaling from the resolution of 240×200 to 8×8.

The saved points would then be scaled down or normalized to a resolution of 8×8 pixels. By scaling it down, it would indirectly make multiple points to be merged into 1 pixel. Hence, making it easier to be compared to a preset value of characters as shown in Figure 7.

2.4.3 Bitwise XOR Operation

This recognition algorithm uses only the XOR operation. The XOR truth table is as shown in Table 2.

Table 2 Truth Table of a XOR logic

INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

To use XOR operation, it requires 2 input which the first input is the handwritten character which had already been converted to 8×8 pixel and the second input is the preset 8×8 character by the system. The preset characters are as shown in Figure 9 and Figure 10.

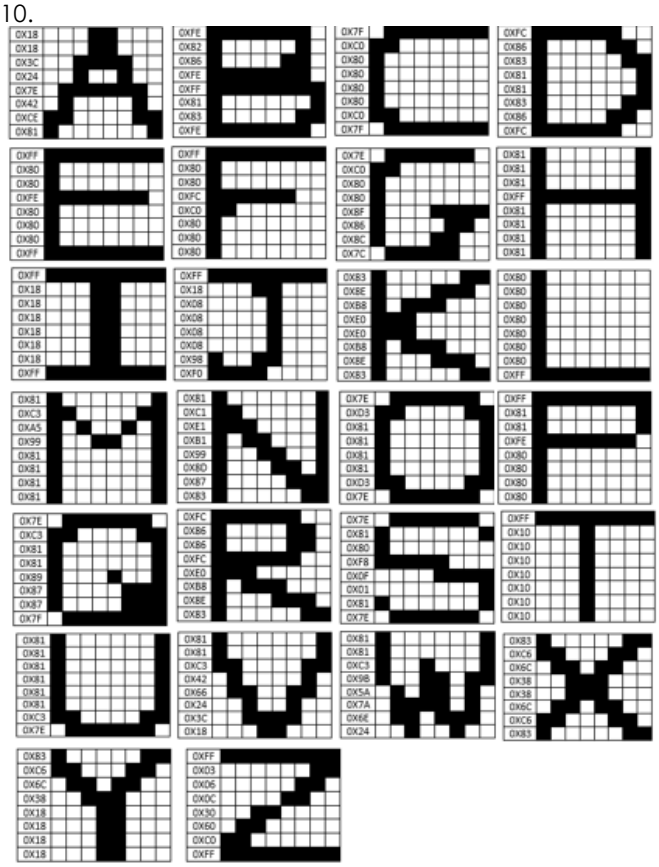


Figure 9 Preset alphabets to be compared in the code

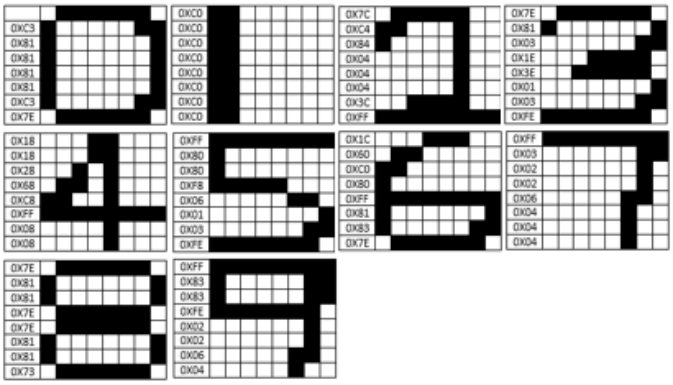


Figure 10 Preset numbers to be compared to the code

By having the preset alphabets and numbers which represent the English character and number, we can now compare each of these preset characters to the handwritten input data by using XOR operation.

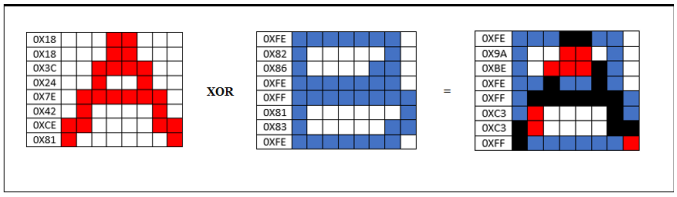


Figure 11 Handwritten Input(colored in red on the left) XOR with (Preset Character 'B' in blue in the middle) resulting the overlap pixels (colored in black on the right)

From Figure 10, we can the red colored pixel on the left are the pixels after letter 'A' was handwritten and rescaled and the middle which is in blue color which resembles the letter 'B' is the preset character in the code. The resulting image of the XOR of the handwritten input and the preset character was shown in the black color pixel in the right image. The black pixel represents the overlapping between the two characters.

As we can see in Figure 11, the overlap pixel between the handwritten character 'A' and the preset character 'B' does not overlap much. So for making this recognition, the total number of pixels were calculated to be 64 pixels which will now be represented as marks. These 64 marks will get reduced by 1 for every pixel that does not overlap. Hence the marks in Figure 10 would be very low since the number of pixels overlapped is very less. But, if the preset character was letter 'A', thus it will overlap even more with the handwritten letter 'A' which will then have very high marks. Thus in that case letter, 'A' would be chosen. In short, the XOR operation is used to determine the number of pixels which do not overlap and the total value of the non-overlap pixels is then taken to reduce the total number of pixels. Hence, the highest number of pixels which overlaps would most likely be the correct recognized character.

3.0 RESULTS AND DISCUSSION

In order to test the functionality of our handwriting recognition system, we had conducted a test in UTeM. The data was collected from 15 volunteer students and analyzed the probability of accuracy for our product. Each of the students was requested to write all the 26 uppercase alphabet and 10 number characters on the touch screen. The results were then recorded in the table in Appendix. From the table in Appendix, we can deduce that:

$$Accuracy = \frac{432}{540} \times 100\% = 80.0\%$$

Based on the data collected from the 15 students, we can conclude that out of 540 tries there were 432 times correct. Thus, the percentage of accuracy for the handwriting recognition system is 80.0%. If we observe the data according to each character, the system can 100% correctly recognized for characters 'A', 'C', 'L', 'U', '0', '2' and '7'. This may due to the unique shape of these characters that make them can be defined and differentiate out easily. On the other hand, the alphabets that had the least accuracy is 'E'. It was correctly recognized by only 6 times out of 15 times, that is only 40% which is less than half. This alphabet was frequently recognized as 'C' or 'F'. However, the alphabet 'C' was perfectly recognized by the system while the error for alphabet 'F' came out was only alphabet 'T'. Hence, this proved that it may not be caused solely by the problem of similar shape but it may due to the recognition in coding. For numbers, the least accuracy was number 8. This may due to the similar shape with number 3. Moreover, this may also due to the handwriting of the user that might not match with the character that defines inside the code. Somehow, there is an increase in the accuracy of the handwriting recognition system after we edit the recognition coding. In addition, the accuracy of the system also increases when the handwriting is neat.

4.0 CONCLUSION

As we can see from the result section, the success recognition rate is 80.0%. The recognition system faces several challenges. For example, when the writing is tilted, it will not be able to recognize the character. Furthermore, every person has a different way in writing English characters, due to that fact there are times where the preset character in the code will differ to some other people's way of writing certain characters. In addition, Arduino Mega has a limited amount of memory. Therefore, at certain times, putting in a larger sample character is not possible.

To further enhance the recognition process, it is possible to make the scaling larger to 10×10 pixel. This may increase the recognition rate as more pixels are used to make the character clearer to be recognized. Another method that could make the system better would be the artificial neural network (ANN) as the

backbone of the recognition process. There are many offline recognition processes that use ANN and has proven to be quite successful. Overall, a portable online handwriting character recognition system with a decent success recognition rate has been successfully developed.

Acknowledgement

Authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) and Ministry of Higher Education for supporting this research under FRGS/1/2015/TK04/FKEKK/02/F00266.

References

[1] Priyanka.M, Ramakrishanan, S., Raajan, N.R. 2013. Electronic Handwriting Character Recognition (E-Hwcr). *International Journal of Engineering Aand Technology (IJET)*. 5(3).

[2] K. P. Ambika, U. Ramesh, K. Saravanan, and J. Hencil Peter. 2013. An Enhanced Version of Pattern Matching Algorithm using Bitwise XOR Operation. *International Journal of Computer Applications*. 68(23): 0975-8887.

[3] Wei, T. J. 2011. *Handwriting Recognition System*. Faculty of Electrical Engineering. Malaysia, Universiti Teknologi Malaysia. Bachelor of Electrical Engineering (Computer). 62.

[4] Rachana R. Herekar, Prof. S. R. Dhotre, 2014. Handwritten Character Recognition Based on Zoning Using Euler Number for English Alphabets and Numerals. *IOSR Journal of Computer Engineering (IOSR-JCE)*. 16(4): 75-88.

[5] Olarik Surinta, Lambert Schomaker and Marco Wiering. 2013. A Comparison of Feature and Pixel-Based Methods for Recognizing Handwritten Bangla Digits. *12th International Conference on Document Analysis and Recognition, 2013.* Washington, DC. 25-28 Aug. 2013. 165-169.

[6] Farha Mendi, Srinivasa.G, Ashwini A.J and Hemanth Krishna H.K. 2013. Online Hand Written Character Recognition. *IOSR Journal of Computer Engineering (IOSR-JCE)*. 11(5): 30-36.

[7] J. Pradeep, E. Srinivasan and S. Himavathi. 2011. Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network. *International Journal of Computer Science & Information Technology (IJCSIT)*. 3(1).

[8] Radzi, S.A. and M. Khalil-Hani. Character Recognition Of License Plate Number Using Convolutional Neural Network. *International Visual Informatics Conference*. 2011. Springer.

[9] Saad, N., *et al.* 2017. Real-Time LCD Digit Recognition System. *Indonesian Journal of Electrical Engineering and Computer Science.* 6(2).

[10] Patil, V. and S. Shimpi. 2011. Handwritten English Character Recognition Using Neural Network. *Elixir Comput Sci Eng*. 41: 5587-5591.

[11] Itqan, K. *et al.* 2016. User Identification System Based On Finger-Vein Patterns Using Convolutional Neural Network. *ARPN Journal of Engineering and Applied Sciences*. 11(5): 3316-3319.

[12] Syafeeza, A., et al. 2015. Convolutional Neural Networks with Fused Layers Applied to Face Recognition. *International Journal of Computational Intelligence and Applications*. 14(03): 1550014.

[13] Liew, S. S., *et al*. 2016. Gender Classification: A Convolutional Neural Network Approach. *Turkish Journal of Electrical Engineering & Computer Sciences*. 24(3): 1248-1264.

[14] Syafeeza, A., *et al.* 2017. Design of Finger-vein Capture Device with Quality Assessment using Arduino Micrcontroller. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC).* 9(1): 55-60.

[15] Lengare, P. S. and M. E. Rane. 2015. Human Hand Tracking Using MATLAB to Control Arduino Based Robotic Arm. *2015 International Conference on Pervasive Computing (ICPC)*. IEEE.

Appendix

Table 3 Result of handwriting recognition done by 15 students

	CHARACTER RECOGNITION																														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9	TTL
S1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	30
S2	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	0	1	29
S3	1	0	1	0	1	1	1	1	1	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	25
S4	1	0	1	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	29
S5	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	30
S6	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	30
S7	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	29
S8	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	1	0	1	27
S9	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	29
S10	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	32
S11	1	1	1	1	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	31
S12	1	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	29
S13	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	1	28
S14	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	30
S15	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	1	1	0	1	0	1	0	0	24
TTL	15	11	15	7	6	14	11	12	13	10	12	15	10	12	13	12	10	8	11	10	15	14	14	14	14	13	15	10	15	11	10	13	11	15	8	13	432

Note: 1 – Character recognized correctly
0 – Character recognized wrongly