ISSN 2278-3091

Volume 8, No.1.1, 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse0981.12019.pdf

https://doi.org/10.30534/ijatcse/2019/0981.12019

Real time License Saudi Plate Recognition Using Raspberry Pi

Loay Alzubaidi[1], Ghazanfar Latif[2]

[1]Department of Computer Engineering & Science, Prince Muhammad bin Fahd University
AL-Khobar, Saudi Arabia
lalzubaidi@pmu.edu.sa

4[2]Department of Information Technology, Prince Muhammad bin Fahd University
AL-Khobar, Saudi Arabia
glatif@pmu.edu.sa

ABSTRACT

With the advent of powerful and low cost Single Board Computers (SoC), developing fully automated systems is becoming a trivial task. We can sense objects and capture images by using embedded systems to process and perform action accordingly. License Plate (LP) recognition systems use optical character recognition techniques to detect number plate on vehicles. These systems are important to control traffic, speed, theft, toll payment and parking lot access for vehicles. In this paper we proposed Arabic LP recognition system which is applicable to the Saudi Arabian LPs with two different formats by using Raspberry Pi board. Therefore, this system recognizes both Arabic and Latin numerals and alphabets. The LP recognition process consists on many preprocessing steps to segment numbers and characters from the captured image of vehicle. The features are extracted based on the horizontal projection profiles and pixel distributions in divided zones of the characters segmented image. Further characters recognition is done for each region of the LP which allows to use SoC with limited processing power. Neural Network and distance classifier are used to recognize different characters in LP. The proposed system correctly recognizes LP with 90.6% accuracy. The low cost fully automated system is easily portable which can be deployed at remote location to control traffic.

Keywords: Single Board Computer, Raspberry Pi, Saudi Licence Plate, Real time Plate Number Plate Recognition, KNN.

1. INTRODUCTION

These days, everything now tends to be moving toward automating. People were used to deal with everything manually, for example, people used to open the gates manually, which means that users had to stop the vehicle, and wait for someone to check their authorization before passing the gate. This process requires at least one man to stand by the gate and check the vehicle, open the gate manually, and then closing it. After the invention of the remote controlled garage doors has caused a great impact on making the lives of the consumer easier; the security person will open and close the gate with a press of button. However, as technology improves the lives of the consumers become easier. Thus, this system is aiming to have the gate to open automatically without needing a person spending his whole day standing to press a button.

The system approaches the same idea in an easy and automated way by recognizing the vehicle's plate number, then if authorized the system will automatically open the gate by using low cost embedded system. One of the biggest advantages of automation is ensuring the quality and consistency of the product without forgetting the important aspect which security. The system is going to automate the functionality of the gate systems by using a unique sign for opening the gates. In other words, each individual vehicle has its own unique plate which is going throw identification and security processes [1].

This research aims to design and the implementation of the Plate Number Recognition system. Unlike the gate's opener that uses a remote control in a hand of human as third party, the system takes picture of a detect approaching vehicle, analyze the images and only opens the gate when a recognized vehicle plate is identified. The main objective of the research is to develop a real time fully automated number plate recognition system that is based on Raspberry Pi. This system will be built using the raspberry as the main component. The system will be able to detect the vehicle, recognize the plate, compare it with the database and control the gate.

2. BACKGROUND

With the start of 20th century, automobile industry boomed and number of motorized vehicles increased rapidly. From 1890 to 1910 the world witnessed a transition from horse to automobiles. As the number increased, law enforcement officials started facing issues to maintain vehicles record and trace them. As a result, in 1890 first number plate was introduced by France and Germany also followed them by introducing in 1993. In United States, Massachusetts was the first state who introduced number plate in 1903 with proper vehicle registration and driver's license registration. Netherland become the first country by introducing national license plate in 1899 by starting license plate with number 1 which reaches to 2001 in 1906 as they selected different way to number the license plates [2]. Figure. 1 shows some of the initial number plates introduced by different countries.

Figure 1: Very first license plate designs in different countries

In 1938, the first oil well was discovered in Saudi Arabia. However, because of World War II in 1939 the Saudi government delayed the development programs and research on the oil industry until 1946. From 1946 to 1950, the Kingdom of Saudi Arabia witnessed a revolution in the oil industry, which raised the country's economy and in this period traffic in Saudi Arabia was on the rise, which led to the development of the licensing plate to register the necessary information regarding automobiles owners. The first license plate in Saudi Arabia appeared in 1950-1962, where they differed from one region to another as shown in Figure 2. In 1972, license plates were established in the entire country with different types of use (privet, bus, taxi and truck) as showing in Figure. 3. However, in 2007 the design was change once again, because license plates were not enough for the demand and population increase which is shown in Figure. 4 [3]. The new version was different from previous ones; the 1996 series was considered to be most preferred by the majority of the public.

Figure 2:1st series of Saudi Arabia license plate

Figure. 3: Unified license plate for the entire country

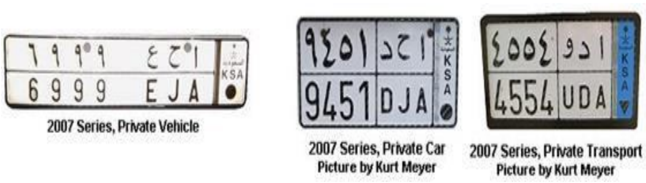


Figure. 4.: New license plate design to meet the increased population demands

3. PROPOSED SYSTEM

The proposed system consists in to different modules. The first module will be dealing with the hardware part of our system. The second module will be dealing with the software part of our system. The third module will be dealing with the database part of the system. The fourth module will be dealing with the design part of our system. Figure. 5 shows flow chart of proposed system. The system's start point is the ultra-sonic sensor. When sensor detects an approached vehicle it turns on the camera to capture image. The capture image is processed to extract vehicle number by applying KNN technique [4]. The extracted results are compared with database in order to decide to open the gate or not.

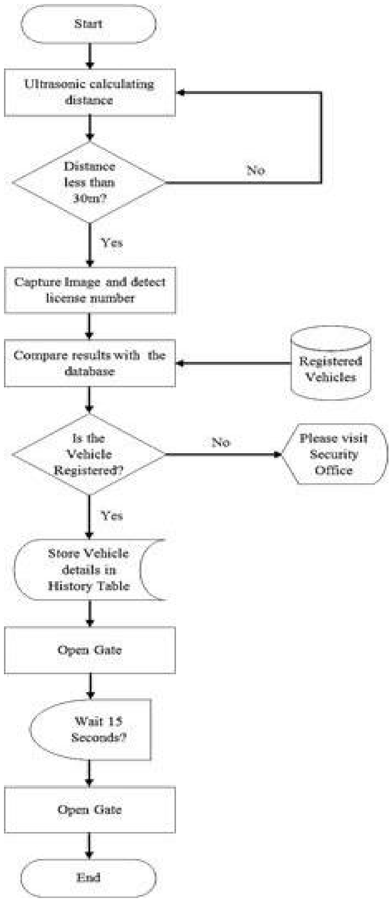


Figure 5: Flow chart of the proposed system

A. Dimensions of Saudi License Plate

The new Saudi license plate standard size is 310mmX155mm (1:2 proportions). The license plate is divided into 5 regions which are shown in Figure. 6. The right part Region 1 (R1) contains name of the country in Arabic; قيدوعسلا, three letters; K S A, and the palm tree of Saudi emblem. The top right Region (R2) has three Arabic alphabets and top left Region (R3) has one to four Arabic numerals. The lower part contains the remaining two regions, R4 has three English letters, R5 has one to 4 numbers.

Region 3 (R3)	Region 2 (R2)	Region 1 (R1)
Region 5 (R5)	Region 4 (R4)	

Figure. 6: Saudi license plate regions

The Saudi plate is on of very few plates in the world that has more than on language which made it harder to design an algorithm that will read them both in a sufficient time. The

Saudi license plate Arabic letters are also mapped with English letters as shown in Table 1.

Table 1: English to Arabic letters mapping

No	Arabic letter	English letter	Description
1	ا	A	***
2	ب	B	***
3	ح	J	Does not have English letter similar to pronunciation of the letter (ج)
4	د	D	***
5	ر	R	***
6	س	S	***
7	ص	X	Letter (S) was served for letter (س)and letter (C) is similar to (G)
8	ط	T	***
9	ع	E	***
10	ق	G	***
11	ك	K	***
12	ل	L	***
13	م	Z	(M) is similar to (N) and is thus, rejected too wide
14	ن	N	***
15	هـ	H	***
16	و	U	(W) is thus, rejected too wide
17	ي	V	(Y) is thus, rejected too high

B. Connecting System Components Togather

For real time fully automated Saudi number plate recognition system, we combined different hardware and software components together. The proposed system mainly relies on Raspberry Pi 3.0 which contains b4 bit quad core 1.2GHz processor, 1GB SDRAM, 440MHz VideoCore IV GPU with 802.11n/Bluetooth wireless support [5]. This model of the Raspberry Pi contains 40 pins including 26 General Purpose Input Output (GPIO) pins and pins for 3V/5V voltage supply [6]. These pins can be controlled by C/Java/Python script to link between hardware and software components along with OpenSource Computer Vision (OpenCV) ibraries[7][8]. High performance cortex Cortex-A53 processor has four processors cores with L2 cache that supports both 64 bit and 32 bit applications with low power consumption. The Raspberry Pi board also has 4 USB ports and an Ethernet port. We used the USB port for our Camera that will used to capture the vehicle image for number plate recognition. The full description of Raspberry Pi 3 board is shown in Figure. 7.

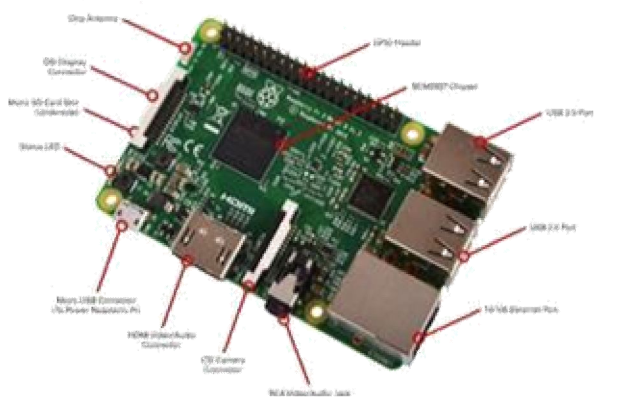


Figure 7: Description of Raspberry Pi 3 Board Components

We used HC SR04 ultrasonic sensor which has 4-Pin circuit board which is shown in Figure. 8. When the Raspberry pi provides the Vcc with 5V then there is a working current of 15mA through the circuit. Using ohms' law then we can figure out that the inner resistance of the entire circuit is 333Ω. When the python code is executed a 5V pulse is sent to the Trig pin to generate a 40 kHz wave from both sensors in the forward direction. The maximum range of these wave pulses are 4m, the minimum range where it can give you a distance is 2cm. The of these waves that are being generated is a 15^0 angle. This is equivalent to $\theta = \frac{15\pi}{180}$, after the wave pulse has been sent, when the pulse hits an object and bounces back to the sensor the trig will become high for 10 µ Seconds (Figure. 9) indicating that there is an object in range. It then shoots 8 cycle bursts of ultrasound at 40 kHz through the echo these 8 cycle bursts are called "Sonic Burst" [9]. The range can be calculated from the moment the trigger signal was sent and the echo signal received by using equation 1.

$$Distance = \frac{High\ Level\ Time \times Velocity\ of\ sound\ \left(\frac{340m}{s}\right)}{2} \quad (1)$$

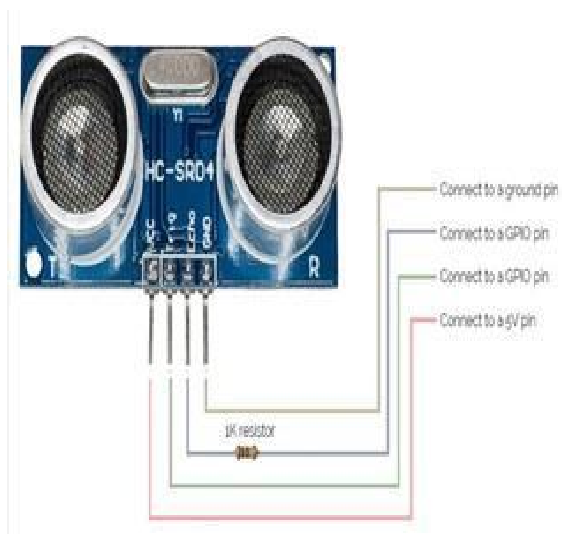


Figure. 8: HC-SR04 ultrasonic sensor pins description

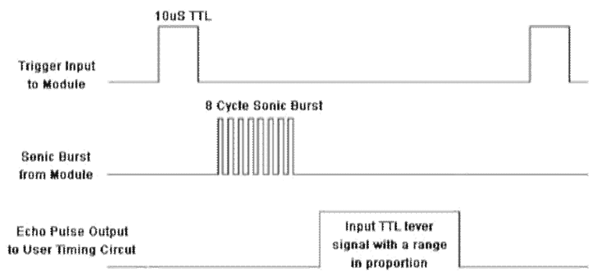


Figure. 9.: Ultrasonic Sensor Timing Diagram

We also used lightweight and small sized SG90 servo motor which can rotate to 180º degrees 90º in each direction. It can rotate with speed of 60º per second and operates with 4.8V to 6 Voltage. We controlled the servo motor by using python library in Raspberry Pi. This Servo motor consists of 3 pins PWM, Vcc and GND which is described in Figure. 10. To control servo motor, we adjusted frequency (or period) and duty cycle to set the servo angle. We look up the timing for our specific servo and Hitec HS-645MG is being used in the example. 0^{o} angle requires a high pulse for 600 us (0.6 ms) and 180^{o} angle requires a 2400 us (2.4ms) pulse. Therefore, to achieve a spread of 180^{o} movement we need a spread of (2.4ms – 0.6ms) 1.8 ms pulse time and we required 0.9ms pulse time. Based on these calculations, we required 0.01ms time pulse per degree.

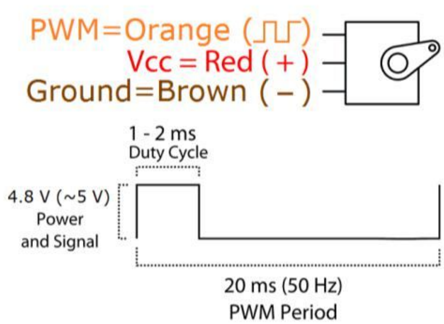


Figure. 10: Servo Motor pins and PWM Cycle

To maintain the servo position, we need to send a pulse every 10ms or we can say we require frequency of 100Hz as in Equation 2. Based on the above time pulse calculations, we can calculate the duty cycle for the desired angle of servo motor as in Equation 3 [10].

$$Required\ Frequency = \frac{1}{0.01} = 100Hz \qquad (2)$$

$$Duty\ Cycle\ in\ \% = \big((Desired\ Angle\ \times 0.01ms) + 0.6ms\big) \times 10ms \qquad (3)$$

For capturing images of the vehicle, we used Logitech c310 camera which needs a supply voltage of 5V from the USB port in addition to 100mA of current giving it according to ohms law an internal resistance of 50Ω. The camera captures an image of 5-megapixel resolution and HD video 1280×720 pixels.

C. License Plate Recognition

The first License Plate Recognition method that we used in our system was based on Tesseract-OCR. Tesseract was originally developed initially in 1994 by the Hewlett Packard (HP) Laboratories in 1994 which further improved in 1998 to support C++ in Windows [11]. In 2005 HP made Tesseract open source. From 2006, onward Google is making changes in it to further enhance it. Currently the OCR support more than 100 languages to recognize them [12].

K-nearest neighbor (KNN) is classifier which is characterized by being instance-based learning and supervised classifier [13]. New instance queries result to the classification of the majority of K-nearest neighbor category. It uses training samples and attributes to classify a new object, and it then determines which are the nearest neighbors of any instance through ‘correlation’ ‘cosine’, ‘hamming’, ‘city block distance’, or ‘Euclidean’ distance. The K-nearest neighbor does this on a ‘random’, ‘nearest’, or ‘consensus’ rule, with ties broken randomly [14]. What makes K-nearest neighbor important is that since its process of classification involves analyzing a small group of objects that are similar, it is found to be very useful for multi-modal classes wherein several objects with independent variables possess varying characteristics on various subsets. It has a record of being accurate even when targeting a class that is multi-modal. On the other hand, this ability to quickly compute for similarities means the K-nearest neighbor treats all features as equal when it computes for similarities. This leads to classification errors and poor measures of similarity, especially when it involves only a small subset of features for classification. It is illustrated in Figure 11 where instance for black-colored objects ɷ=3 results to grouping together the black-colored objects with red-colored objects and classifying them as belonging into the same class.

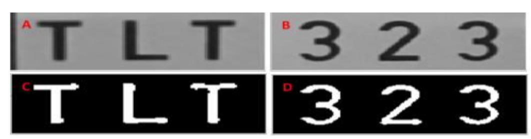
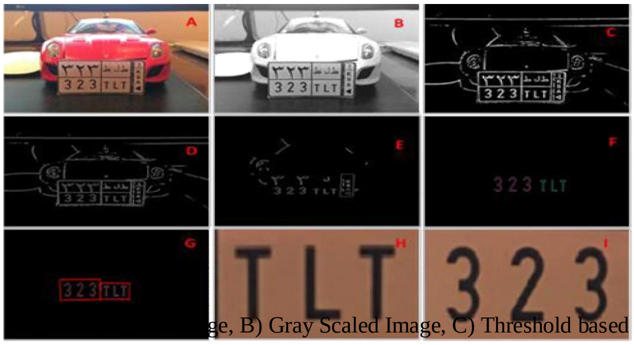


Figure. 11: Converting from RGB to Gray and then Binary image of detected license plate

With the same concept, we used KNN as an algorithm for character detection, the algorithm needs to be trained first for a certain set of characters then it became ready to use and compare what it sees with what It has been trained on. Understanding the concept of KNN is not enough to implement it in real case, since the input image won’t be as clear as the algorithm would like it to be, we need set of image processing steps that will prepare the image for extracting information in it and then lock for the suitable matches and assess each one of them to see whether it satisfy being a character or not [15]. The process is mainly two parts; the first is locating the plate of the image then detecting the characters in the plate itself using k-NN. If the first part of the process failed to successfully locate a Plate, the whole process is failed. Before passing captured image to the Tesseract, we applied preprocessing techniques including converting colored image to gray level, erosion and dilation [16][17]. The sample results of the extracted plates are shown in Figure 12.



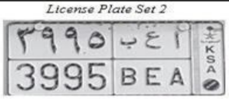
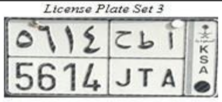
ge, B) Gray Scaled Image, C) Threshold based

Figure 12:. Binary Image, D) Image after finding all Contours, E) Image after finding possible Characters, F) Image after finding all vectors of matching Characters, G) Boundary of matching Characters of plate part, H) Extracted English letters part of the plate, I) Extracted numbers part of the plate

4. EXPIREMENTAL RESULTS

Table 2 shows a general comparison between all the three algorithms we used and how accurate the result is, generally KNN gives the most accurate result due to our modification and implementation of it. Based the achieved results we can see that the license plate recognition method that had the most accurate results is the method based on KNN. KNN resulted in recognizing the previous tested images with an average of 90%. Whereas the OpenALPR the license plate recognition method resulted in recognizing the tested images with an average of 75%. Furthermore, the Tesseract-OCR based license plate recognition method resulted in recognizing the tested images with an average of 55%. After looking to these results, we decided to implement the KNN based license plate recognition method as it resulted with the highest percentage of accuracy of 90%. Figure 13 shows the performance comparison of different LP detection methods.

Table 2: Performance of different methods

Plate/Method	*Tesseract*	*OpenALPR*	*KNN*
License Plate Set 1 	40%	70%	90%
License Plate Set 2 	60%	80%	92%
License Plate Set 3 	61%	74%	95%
License Plate Set 4 	57%	75%	88%
License Plate Set 5 	48%	70%	88%
Average	**53.20%**	**73.80%**	**90.60%**

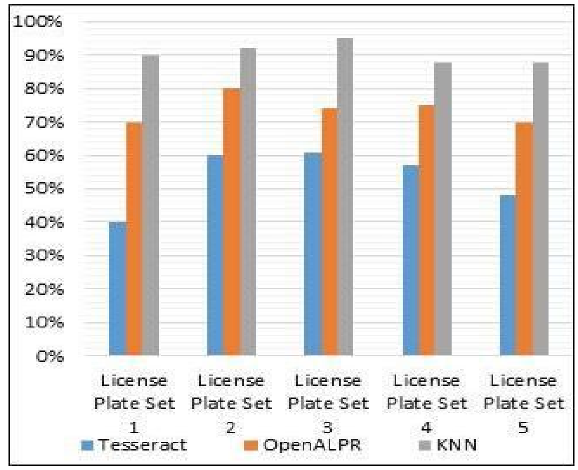


Figure. 13: Performance comparison chart of different LP detection methods

5. CONCLUSION

Our research aims to create integrated systems that will reduce man labor, discard redundant work and to create an automated future. We have summed up three different ways to process the license plate OpenALPR, Tesseract and KNN. We showed the different results of each algorithm singling out the KNN for its superior results in terms of license recognition. The ultrasonic measures the distance of the car approaching the gate, when a certain distance is measured an instruction will be sent to the camera to capture a picture of the car's license plate. This image gets processed and runs as input to the KNN algorithm, opening the gate if the result is found in the database, otherwise the gate will not open. This system can be integrated into main gate substituting the need for security personnel to be stationed there all the time. When a vehicle is verified by the security official, its license plate details are being inserted to the database. These information from the database are used to open the gate once the license plate is verified, making it easier for the security personnel to make their rounds and focus on other useful things rather than stay at the gate and open it manually all the time.

In future, we will try to improve the algorithm to recognize Arabic Letters and numbers. We will also try to add a library to the software with all GCC License plates. We will also add more training and testing data to improve the results. In hardware level, we will add LCD to display the important messages to the system users as well as will use different LED lights to indicate that a vehicle is allowed or denied to enter.

REFERENCES

[1] Arth, C., Limberger, F., & Bischof, H. "Real-time license plate recognition on an embedded DSP-platform," IEEE Conference on Computer Vision and Pattern Recognition , pp. 1-8, 2007. https://doi.org/10.1109/CVPR.2007.383412

[2] Kothman, G. S., "U.S. Patent No. D163,328," U.S. Patent and Trademark Office,1951.

[3] Saudi Arabian Private and Passenger vehicle license plate History: http://www.worldlicenseplates.com/ world/AS_SAUD.html

[4] Boiman, O., Shechtman, E., & Irani, M. "In defense of nearest-neighbor based image classification," In Computer Vision and Pattern Recognition, IEEE Conference, pp. 1-8, 2008. https://doi.org/10.1109/CVPR.2008.4587598

[5] Monk, S. "Raspberry Pi cookbook: Software and hardware problems and solutions," O'Reilly Media, 2016.

[6] Rasp Richardson, M., & Wallace, S., "Getting started with raspberry PI," O'Reilly Media, 2012.

[7] Monk, S., "Programming the Raspberry Pi: getting started with Python," McGraw Hill Professional, 2015.

[8] Brahmbhatt, S., "Embedded Computer Vision: Running OpenCV Programs on the Raspberry Pi," In Practical OpenCV, pp. 201-218, 2013.

[9] Carullo, A., & Parvis, M., "An ultrasonic sensor for distance measurement in automotive applications," IEEE Sensors journal, pp, 143-147, 2001.

[10] Dote, Y., "Servo motor and motion control using digital signal processors," Prentice-Hall, 1990.

[11] Smith, R. W., "History of the Tesseract OCR engine: what worked and what didn't," In IS&T/SPIE Electronic Imaging, pp. 865802-865802, 2013.

[12] Patel, C., Patel, A., & Patel, D., "Optical character recognition by open source OCR tool tesseract: A case study," International Journal of Computer Applications, 2012.

[13] Keller, J. M., Gray, M. R., & Givens, J. A., "A fuzzy k-nearest neighbor algorithm," IEEE transactions, vol. 4, pp. 580-585, 1985.

[14] Mitchell, Tom M., "Bayesian Learning, In Machine Learning," McGraw Hill, pp. 154–178, 1997.

[15] Kosbatwar, S. P. and Pathan, S. K., "Pattern Association for character recognition by Back- Propagation algorithm using Neural Network approach," International Journal of Computer Science & Engineering Survey, vol. 3 – no. 1, pp. 127-134, 2012.

[16] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. https://doi.org/10.1098/rsta.1955.0005

[17] Chen, C. W., Luo, J., & Parker, K. J.," Image segmentation via adaptive K-mean clustering and knowledge-based morphological operations with biomedical applications," IEEE Transactions on Image Processing,vol. 7, no. 12, pp.1673-1683,1998. https://doi.org/10.1109/83.730379

[18] Patel, C. I., Patel, R. and Patel, P., "Handwritten Character Recognition using Neural Network," International Journal of Scientific & Engineering Research, vol. 2 – no. 5, pp. 1-6, 2011.