Affordable and Portable Realtime Saudi License Plate Recognition using SoC

Loay Alzubaidi
Department of Computer Science,
Prince Mohammad bin Fahd
University,Al Khobar, Saudi Arabia.
lalzubaidi@pmu.edu.sa

Ghazanfar Latif
Department of Computer Science,
Prince Mohammad bin Fahd
University,Al Khobar, Saudi Arabia.
glatif@pmu.edu.sa

Jaafar Alghazo
Department of Computer Engineering,
Prince Mohammad bin Fahd
University,Al Khobar, Saudi Arabia.
jghazo@pmu.edu.sa

***Abstract*— Stand along single board computers (SoC) have become so inexpensive and yet so powerful that paved the way for easily developing fully automated systems. SoC systems are equipped with sensors, cameras and various embedded systems that allow developing systems that interact with the surrounding environment. Therefore, the task of capturing images of License plates and using Optical Character Recognition (OCR) techniques to recognize the numerals and characters allows for developing an inexpensive License Plate (LP) Recognition system. LP systems are important and can be used for various application from traffic control, toll payment, and parking access. This paper proposes a Raspberry PI based LP recognition for Arabic/English Characters and Numeral on license plates used in Saudi Arabia. The proposed process utilizes the phases of Preprocessing, Segmentation, Feature Extraction and Classification. At the end of the preprocessing phase, the Characters and Numerals are segmented. Pixel distribution and Horizontal projection profiles are used in the feature extraction phase for the segmented image. Distance Classifier and k-nearest neighbors classifier are used in the classification phase. The proposed system achieved an accuracy of 90.6%. The advantage of such a system is the low cost and portability making it affordable and easily deployable in any location.**

Keywords*— *Single Board Computer; Raspberry Pi; Saudi License Plate; Real time Plate Number Plate Recognition; KNN

I. 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, analyse 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.

II. 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]. Fig. 1 shows some of the initial number plates introduced by 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 Fig. 2. In 1972, license plates were established in the entire country with different types of use (privet, bus, taxi and truck) as showing in Fig. 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 Fig. 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.

978-1-7281-2882-5/19/$31.00 ©2019 IEEE

Fig. 1. Very first license plate designs in different countries

Fig. 2. First series of Saudi Arabia license plate

Fig. 3. Unified license plate for the entire country

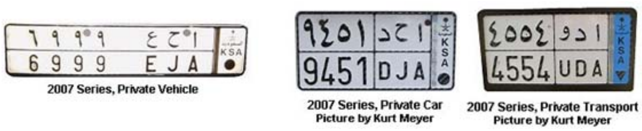


Fig. 4. New license plate design to meet the increased population demands

III. 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. Fig. 5 shows flow chart of proposed system. The system's start point is the ultrasonic 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.

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 Fig. 6. The right part Region 1 (R1) contains name of the country in Arabic; الســـعودية, 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.

B. Connecting System Components Together

For real time fully automated Saudi number plate recognition system, different hardware and software components are combined together. The proposed system mainly relies on Raspberry Pi 3.0 which contains b4 bit quad core 1.2GHz processor, 1GB SDRAM, 440MHz Video Core 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 Open Source Computer Vision (OpenCV) libraries [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. The USB port is used 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 Fig. 7.

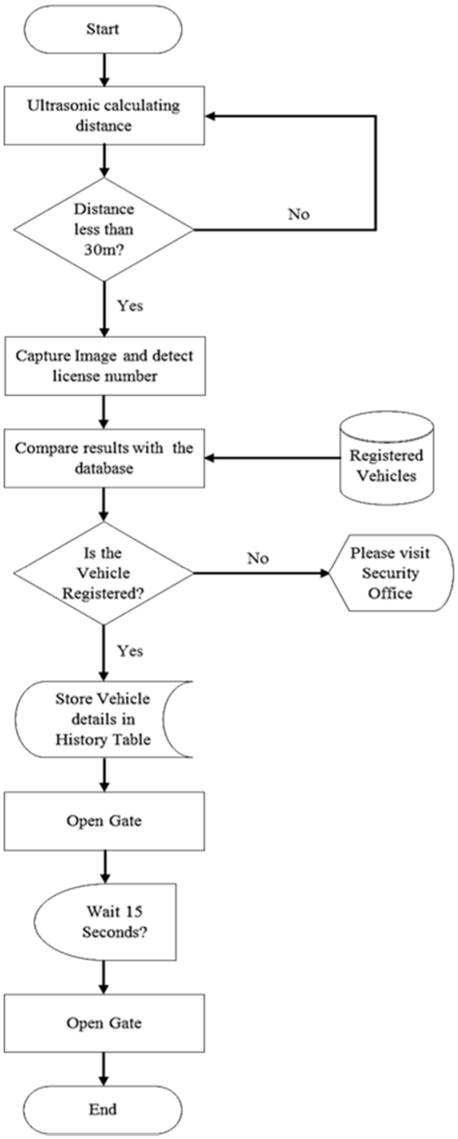


Fig. 5: Flow chart of the proposed system

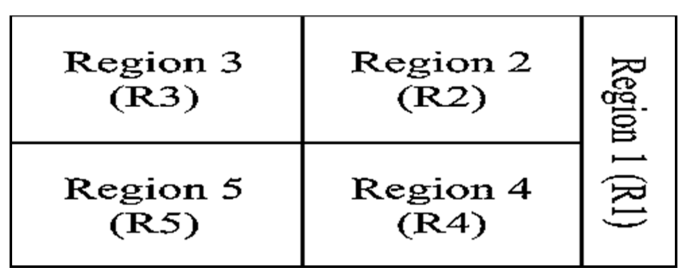


Fig. 6: Saudi license plate regions

An HC-SR04 ultrasonic sensor is used which has 4-Pin circuit board which is shown in Fig. 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 it 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 θ of these waves that are being generated is a 15□ 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 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 which is shown in Fig. 9.

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

Lightweight and small sized SG90 servo motor is used 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. The servo motor is controlled by using python library in Raspberry Pi. This Servo motor consists of 3 pins PWM, Vcc and GND which is described in Fig. 10. To control servo motor, frequency (or period) is adjusted 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. 0º angle requires a high pulse for 600 us (0.6 ms) and 180º angle requires a 2400 us (2.4ms) pulse. Therefore, to achieve a spread of 180º movement, a spread of (2.4ms – 0.6ms) 1.8 ms and 0.9ms pulse time is required. Based on these calculations, 0.01ms time pulse per degree is required.

To maintain the servo position, need to send a pulse every 10ms or required frequency of 100Hz as in Equation 2. Based on the above time pulse calculations, the duty cycle for the desired angle of servo motor is calculated 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.

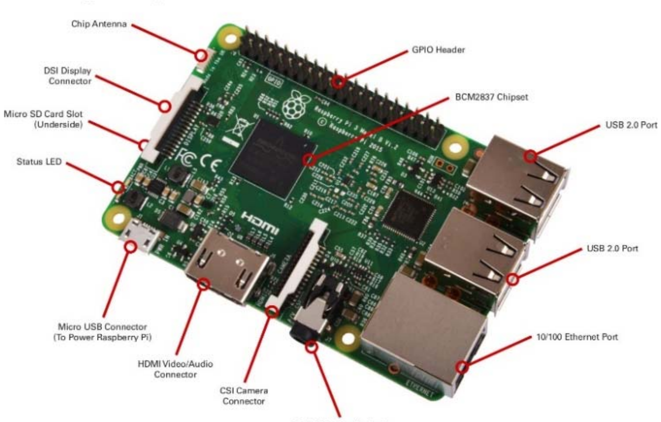


Fig. 7: Description of Raspberry Pi 3 Board Components

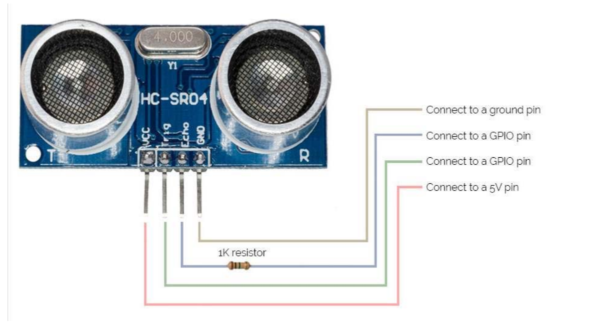


Fig. 8: HC-SR04 ultrasonic sensor pins description

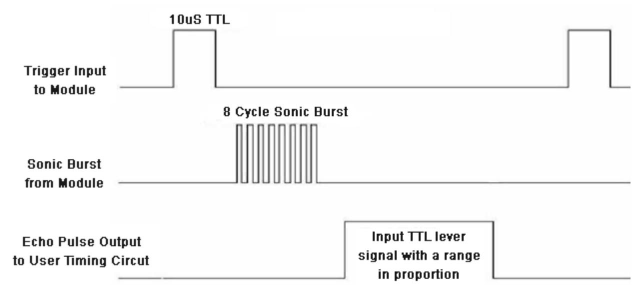


Fig. 9: Ultrasonic Sensor Timing Diagram

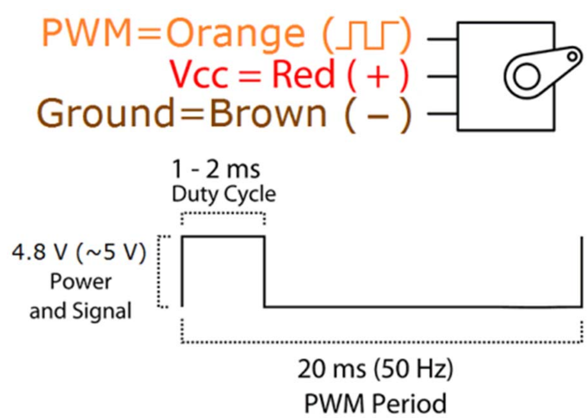


Fig. 10: Servo Motor pins and PWM Cycle

C. License Plate Recognition

The first License Plate Recognition method that used in the proposed 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 a supervised classifier with the ability for instant based learning [13]. The use of training samples along with attribute is used for classifying a new object and subsequently determining the nearest neighbor of any instance through the use of various algorithms [14]. Classification in KNN requires analyzing similar groups. KNN works very good with Multi-Modal classes and is known to be an accurate process. However, in KNN all features are treated equally when computing for similarities. This may lead to classification errors especially when the feature set is small.

TABLE I. 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 (ح)
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

With the same concept, KNN as an algorithm for character detection is used. 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 so needed a 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 KNN. 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, preprocessing is done including converting the color image to grey level, erosion and dilation [16][17]. The sample results of the extracted plates are shown in Fig. 11 and Fig. 12.

IV. 4. EXPERIMENTAL RESULTS

Table 2 shows a general comparison between all the three algorithms used and how accurate the result is, generally KNN gives the most accurate result due to our modification and implementation of it. Based on 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, it is 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 accuracies comparisons of proposed method with the other existing techniques.

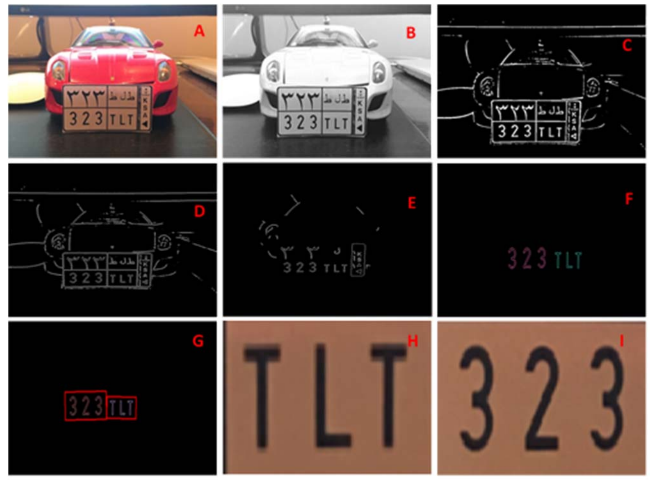
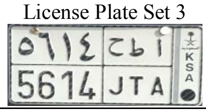


Fig. 11: A) Original Image, B) Gray Scaled Image, C) Threshold based Binary Image, D) Image after finding all Contours, E) Image after finding possible Characters, F) Image after finding all vectors of matching Characters, G) Boundary of matching Characters of plate part, H) Extracted English letters part of the plate, I) Extracted numbers part of the plate.

TABLE II. 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	**53.2 %**	**73.8 %**	**90.6%**

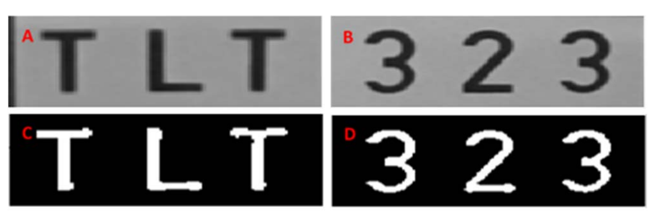


Fig. 12: Converting from RGB to Gray and then Binary image of detected license plate.

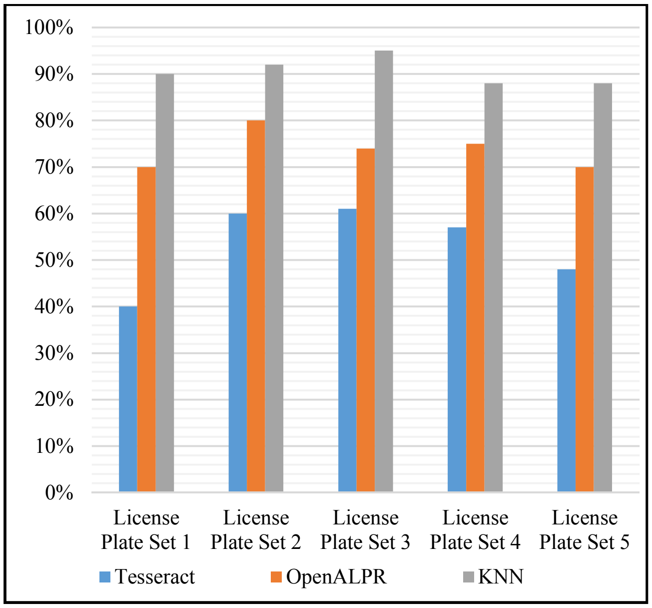


Fig. 13: Performance comparison chart of different LP detection methods

V. CONCLUSION

Our research aims to create integrated systems that will reduce man labor, discard redundant work and to create an automated future. Three different ways to process the license plate OpenALPR, Tesseract and KNN are discussed. 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 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. (2007, June). Real-time license plate recognition on an embedded DSP-platform. In 2007 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE..

[2] Kothman, G. S. (1951). U.S. Patent No. D163,328. Washington, DC: U.S. Patent and Trademark Office.

[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. (2008, June). In defense of nearest-neighbor based image classification. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.

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

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

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

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

[9] Carullo, A., & Parvis, M. (2001). An ultrasonic sensor for distance measurement in automotive applications. IEEE Sensors journal, 1(2), 143-147.

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

[11] Smith, R. W. (2013, February). History of the Tesseract OCR engine: what worked and what didn't. In IS&T/SPIE Electronic Imaging (pp. 865802-865802). International Society for Optics and Photonics.

[12] Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. International Journal of Computer Applications, 55(10).

[13] Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, (4), 580-585

[14] Mitchell, Tom M. (1997). Bayesian Learning. In Machine Learning (pp. 154–178). Mitchell, T. (1997). Machine Learning.

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

[16] Chen, C. W., Luo, J., & Parker, K. J. (1998). Image segmentation via adaptive K-mean clustering and knowledge-based morphological operations with biomedical applications. IEEE Transactions on Image Processing, 7(12), 1673-1683.

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