Multidisciplinary Review Journal

Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 3, 2020

Review Article

LICENSE PLATE NUMBER DETECTION AND RECOGNITION USING SIMPLIFIED LINEAR-MODEL

YOUNG JUNG CHOONG[1], LEE KIM KEONG[2], TAN CHYE CHEAH[3]

[1]School of Computing, Asia Pacific University of Technology and Innovation. jungchoong_1212@hotmail.com
[2]School of Computing, Asia Pacific University of Technology and Innovation. kimlee@staffemail.apu.edu.my
[3]School of Computing, Asia Pacific University of Technology and Innovation. chyecheah.t@staffemail.apu.edu.my

Received: 11.11.2019 *Revised: 13.12.2019* *Accepted: 15.01.2020*

Abstract
License Plate Recognition (LPR) is one of the technologies commonly applied in Intelligent Transport System (ITS) nowadays. This technology enables the software system to recognize the license plate number by itself from a digital image. The input image is then converted to a meaningful ASCII text which contains the license plate number. This paper discusses the series of method to recognize the license plate number. The methods include grayscale which turns the colorful image into grayscale, binarization which further converts the grayscale image into the black and white version, license plate detection which is to search for location of the license plate, character segmentation which separate the extracted characters individually and character recognition to transform the pixel into meaningful information. The system has demonstrated more than 90% success rate. In addition, the performance issue has addressed by omitting some pre-processing such as contrast enhancement, noise filtering, and histogram equalization.
Keywords: Grayscale, Image Binarization, Character Segmentation, Character Recognition, Window Sliding, Connected Component Analysis (CCA), Template Matching.

DOI: http://dx.doi.org/10.31838/jcr.07.03.09

INTRODUCTION

License Plate Recognition (LPR) has been a technique used in various Intelligent Transport System (ITS) over the last few decades. It replaced human efforts to recognize the license plate. Instead of high-speed vehicle plate recognition environment, this proposed system with the LPR technology is designed primarily in the scope of a car park area with a camera stationed statically. In a typical LPR system, the first step is capturing the vehicle image with the license plate from the camera. The digital image is then processed to some useful information by applying some image processing techniques. After obtaining the image, the grayscale features take place to the input image which is to find the average color value from the three main component color which are red, green and blue (RGB). After that, the grayscale image is further processed to the binarization and transform the grayscale image to only black and white for future use. With the image in black and white, the system performs license plate detection by searching through rectangle made up with all four white edges in the image which possibly contains the characters within. This step is crucial as the characters may not be detected when the license plate portion is incorrect. After cropping out the detected license plate, the system segments the characters into individual for the recognition purpose by using Connected Component Analysis (CCA). Following that, the segmented character is undergone through a process known as template matching one by one to find the most similar character from the pre-defined character template.

License Plate Recognition (LPR) has been a technique used in various Intelligent Transport System (ITS) over the last few decades. It replaced human efforts to recognize the license plate. Instead of high-speed vehicle plate recognition environment, this proposed system with the LPR technology is designed primarily in the scope of a car park area with a camera stationed statically. In a typical LPR system, the first step is capturing the vehicle image with the license plate from the camera. The digital image is then processed to some useful information by applying some image processing techniques. After obtaining the image, the grayscale features take place to the input image which is to find the average color value from the three main component color which are red, green and blue (RGB). After that, the grayscale image is further processed to the binarization and transform the grayscale image to only black and white for future use. With the image in black and white, the system performs license plate detection by searching through rectangle made up with all four white edges in the image which possibly contains the characters within. This step is crucial as the characters may not be detected when the license plate portion is incorrect. After cropping out the detected license plate, the system segments the characters into individual for the recognition purpose by using Connected Component Analysis (CCA). Following that, the segmented character is undergone through a process known as template matching one by one to find the most similar character from the pre-defined character template.

LITERATURE-REVIEW

There are several techniques have been reviewed into different stage of the entire LPR process. Each stage has its own algorithm to achieve the goal respectively. A brief description of each stage and the working environment behind is covered in this section. The accuracy of the result very much dependent on the resolution i.e., pixels of the image capture, the higher the resolution, the higher the accuracy of the result.

Image Acquisition

Image acquisition refers to the capturing of the number plate by using a camera. However, the image is not just only the number plate when capturing but some background and some unnecessary items. Therefore, S.Kramthi, et al. [7] suggested turning the original colourful image into grayscale as colour is not necessary in this section. Grayscale conversion is the process of searching the average colour by multiplying the number of RGB of the particular pixel with a specified equation which is 0.114 *R+0.587*G+0.299*B [2]. Different colour components are having their own number to be multiplied to achieve the best result when turning into grayscale. Turning the input image into grayscale format will also help the image binarization later. Binarization is the process of applying a threshold value to change the grey value having the range from 0 (most black) to 255 (most white) to the binary value which contain only 0 or 1 [16]. Taking a threshold value 80 as example, if the pixel grey value is below 80, then the pixel will be set with 0, if the pixel value is greater than 80, then the value 1 will be set to that pixel. By doing this, it is easier to differentiate the foreground and also background of an image for the purpose of recognition later. Figure 2.1 presents an example of turning a colour image into grayscale then finally become a binarized image.

Figure 2.1: Colour to grayscale to binarized

License Plate Localization

The localization of the number plate from the entire image can be performed by the Sobel mask. Sobel mask is commonly used for edge detection in image processing. It defines all the edges in the input image. However, X et al. [15] suggested a few pre-processing steps to perform before defining the location of the license plate. These pre-processing steps are contrast enhancement and noise filtering in which help improve the contrast of the uploaded image and also reduce the noise background, respectively. By doing this, the location of the license plate is more visible for the computer to recognize. To extract only the license plate and remove the other unnecessary items or background, the horizontal localization phase responsible for identifying the horizontal segmentation of the license plate, same goes with the vertical localization phase [12]. Sobel Operator works well with convolution masks that help to find out the sharp change of intensity between pixels with a gradient matrix shown in Figure 2.2 [9]. The grey value of the 3x3 matrix pixel is multiplied respectively with the gradient value, then they are sum-up. There is no edge if the result is 0 since there is no high intensity of changes among the pixels.

$$S_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$$

$$S_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

Figure 2.2: 3x3 pixel matrix

However, a limitation of Sobel mask edge detection method for recognizing the license plate is observed especially in case the background colour of license plate is almost the same to the vehicle body colour i.e., black as shown in Figure 2.3.

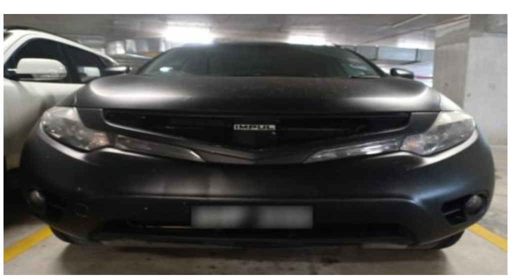


Figure 2.3: Same colour in license plate and vehicle

It is difficult for human eyes to recognize the license plate. With the Sobel mask edge detection technique on this image (Figure 2.3), edge detection can be difficult or may be invisible as there is no intense change between the pixel of the car as well as the license plate. Therefore, a window sliding technique can be used to perform Hough Transform [6] to find imperfect edges of license plate. Hough Transform is the process of finding the possible shape (rectangle) within an image while window sliding method is used for scanning through the entire picture pixel by pixel. Precisely, the Hough Transform can discover all possible rectangle shape all over the image resulting in thousands of rectangle shapes produced. As such, a filtering process is needed to iterate each rectangle for license plate. Thus, the window sliding method with binarization is used to streamline the process.

Most of the car plate registered in Malaysia is having white font with black background for human eye easier to recognize [10]. This means that the almost black pixel will be completely black, and the rest will become all white after the binarization. The binarized image also allows the computer to recognize the license plate quickly. To detect the location of license plate, the system will only have to go for the rectangle which is completely black on all the edges. The system scans through the image with the predefined and modifiable size rectangle, once the rectangle hits the white pixel, then it goes for the next pixel and try to match again until all of the four edges is connected with only black pixel. On the other hand, if the recognition process is run through every pixel of the input image, it requires a lot of computation time for the computer to find out the possible plate. Therefore, a special configuration is made by setting the algorithm where the starting point will be the second-half of the input image as the camera is standstill for image capturing, the license plate is always appeared second-half of the image. This leads 50% of computation time-saving as well as eliminating the possible rectangle at the first-half of the input image.

Character Segmentation

After cropping the license plate, the next step is character segmentation. Character segmentation is to separate the alphanumeric character on the license plate individually. Thus, the characters are needed to be transformed to an array of numerical data. It can be achieved by using the Vertical Projection Profile (VPP), said by [3]. VPP is to determine the brightest colour and also the darkest which match the normal colour of a number plate. In this way, the computer can know the gap between each character and separate accordingly.

However, Patel, et al. [6] and V. Karthikeyan et al. [14] have both suggested another method for license plate segmentation by using Connected Component Analysis (CCA). CCA is referring to the Connected Component Labelling (CCL) which label all the connected pixel in the input image. The computer first scan through only the black pixels of the input image, once the black pixel is found, the computer then check whether its neighbour has been labelled before. If its neighbour has been labelled, then the current pixel will be labelled with the same number with its neighbour. If not, then the current pixel will be assigned with a new label value. Same process will be gone through the cropped license plate and finally separate the characters according how many labels has been created.

Regarding the mentioned CCL, there are two-type of connectivity which are 4-adjacency and 8-adjacency of pixels connectivity [10]. The difference between pixels connectivity is the 4-adjacency only look for the four direction which are east, west, south, north while the 8-adjacency look for southeast, southwest, northeast and northwest. Not only that, 8-adjacency provide the better accuracy than 4-adjacency but requires more computational time. The graphical illustration of the pixel connectivity is shown as Figure 2.4.

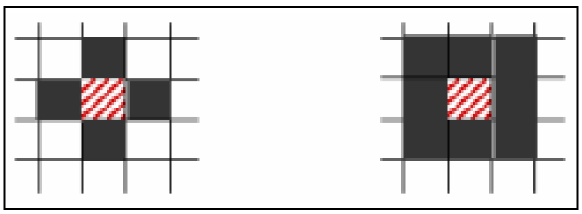


Figure 2.4: 4-neghbourhood and 8-neighbourhood

By applying CCA into the character segmentation, there is one disadvantage which is the system will recognize the item or object which does not belong to any license plate region [13]. Hence, an extra method to filter out the non-license plate region. For instance, the name of vehicle registration company at the bottom of the license plate as well as some stickers usually attached beside the last character. The extra filter criteria might include the height and width of the segmented result. The segmented object must have the height over a threshold value then only can be considered as a character. After extracting the character individually, for each labelled object, the border is defined by searching through the outmost pixel from up, down, left and right. After that, the segmented character needed to be normalized into same size for the future recognition purpose.

Character Recognition

Character recognition is the last step of the entire image processing process. For a computer to recognize each and every character, the first thing to do is training. The technique, template matching will tell the computer what a character looks like [8]. For example, the character 'A' is sharp on top, having two legs at the bottom and one line at the centre. When recognizing, the computer will perform the matching on a pixel-by-pixel basis. Mohammad, et al. [5] suggested to convert the segmented character into binarized array shown in Figure 2.5.

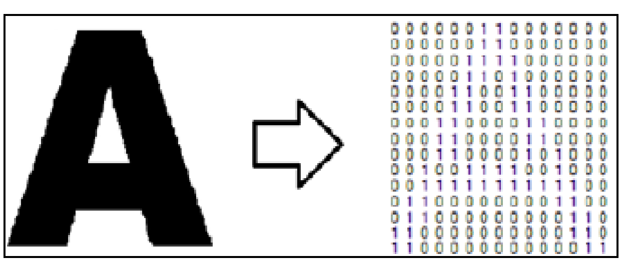


Figure 2.5: Template Matching

In Figure 2.5, the black pixels are transformed into 1 and the white pixels are all labelled as 0. By doing this, the system is easier to understand and figure out the shape of character instead of recognizing the grayscale colour of the image. Then, a pre-defined character image template is prepared to determine the similarities between them later [1]. The similarities will be determined according to the pixel value that has been set, in this case, either 1 or 0 only.

In order to find the similarities of the segmented character image with the template image, a method which is known as Sum of Absolute Difference (SAD) is raised [4] [11]. In the SAD, it computes the absolute difference of all the pixel value respectively taken from two matrix which are having the same size. The first pixel in the segmented image will be compared with the first pixel in the template image, second is with the second and so on, until every single pixel in the image has been compared. As the matrix contains only 0 and 1, therefore the comparison is to check whether the pixels are holding the identical value. If yes, then the score will be increased by 1, if no, the score is remained. One segmented character will be compared with 10 digits and 26 alphabets, and at the end, the character with the highest score will be the recognized character.

SYSTEM ARCHITECTURE

There is a total of five stages separated class-by-class in this solution model which are grayscale, image binarization, license plate detection, character segmentation and character recognition. The process is shown in Figure 3.1 below. The stages are linear and only travel with one direction. No iteration is allowed.

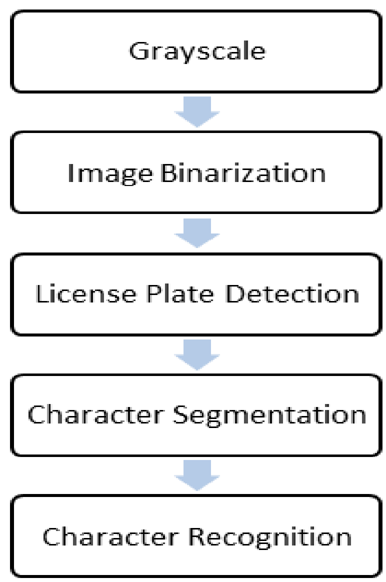


Figure 3.1: Process Model

Grayscale

To perform image binarization, the input image is first converted into grayscale format. To turn a colourful image to grayscale, the average value of Red, Green, and Blue (RGB) is calculated to find out what is the colour value of RGB after the three main colour value has been multiplied with some predefined number. After that, the system is able to find out the grayscale value for each and every pixel in the third quadrant which the range is between 0 (most black) to 255 (most white).

Image Binarization

Image binarization can be taken as the first step of the entire license plate recognition process. This feature allows the system to take the input image and turn it into binary format which contain only white and black pixel as shown in Figure 3.2. The reason for binarization is to ease the system to recognize and differentiate the foreground and background of the image. Not only that, as the colour of the license plate (black) has the high contrast with the colour of the character (white) on it, therefore the binarization process will separate and redefine them. However, only the possible license plate region will be binarized due to the performance concern about the unnecessary or unwanted region, which is about at the $\frac{3}{4}$ third quadrant of the image. A threshold value is pre-defined to tell the system to change the pixel to white colour if the pixel value is less than the threshold value and vice versa.

Figure 3.2: Binarized Image

License Plate Detection

The system is able to detect license plate by searching through the white area from the binarized image. To detect the Region of Interest (ROI), the system will try to find a rectangle which has the four edges connected with only white pixel and contain some black pixel in the middle. This match the characteristic of a license plate. A predefined size of 267 in width and 61 in height is used to form the rectangle to detect the license plate. This is the size of the smallest license plate can be after some observation. On the other hand, not every vehicle is having the same license plate size all the time, therefore the system will increase the estimated width and height accordingly if the system cannot find the ROI license plate at first. After the system has increased the size maximum 3 times, the system will assume the license plate is in square and redefine the width and height in order to find out the ROI which might be the license plate. Figure 3.3 shows the detected license plate region draw in a red colour rectangle.

Character Segmentation

Furthermore, the system provides the capability to segment characters individually from the cropped license plate. The system uses Connected Component Analysis (CCA) to differentiate the character because every character is made up by a group of connected pixels. As the cropped license plate is white colour in background and black colour in character, so the system will focus on finding only the black pixel in the image. Once the system found black pixel, it first looks to the four neighbours around covering four directions, east, south, west, north and see whether they have been labelled. If none of them is labelled, then the system will put the pixel in a list. The list contains width of the pixel, height of the pixel, and a label number. If one of them is found labelled before, then the current pixel will be put into the list as well with the same label number. However, this may cause a conflict where the north and west pixel are having the different label number. In this case, the current pixel will follow the smallest label number while all of the pixels with the larger label number will be updated to the smallest label number as well. Hence, they all become an object as one. Figure 3.4 shows the segmented and resized character.

Figure 3.3: Detected license plate with red rectangle

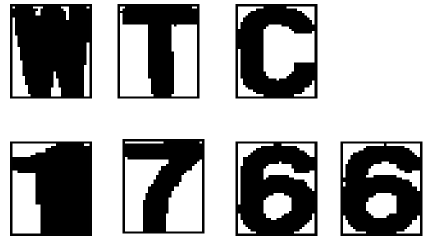


Figure 3.4: Segmented Characters

Character Recognition

The last feature of this solution is character recognition. The system uses template matching as the technique to recognize character. The character template including number 0-9 and alphabet A-Z are stored into an image folder in the system. The size of the character template is fixed, 30px of width and 40px of height. The segmented character is also resized to the same so that it can be matched with the character template. The system will turn both the segmented character and the character template into an array. By using array, the pixel is compared with the colour. As the image contain only black and white pixel, there is a difference if the colour of pixel from both arrays is not same. If the pixel from array1 is black and the pixel from array2 is black as well, the system will consider it as a match and add a score to it. This process will be continued until every pixel in the array is scanned. At the end, the template obtains the highest match area is the recognize character.

TESTING RESULTS

To test the performance of entire recognition process, a total of 30 images has been used and tested from the beginning of grayscale to the end of character recognition. The system is outperformed in which a maximum of 5 seconds it takes to recognize the character. Table 4.1 shows the experimental result of each phases for the entire system about the accuracy and success rate.

Table 4.1: Algorithm Success Rate

Process	Success Rate
Grayscale	100%
Image Binarization	100%
License Plate Detection	89%
Character Segmentation	97%
Character Recognition	98%

For every image, grayscale has the 100% success rate as the algorithm for it is very simple. This is because the system is just getting the red, green, blue (RGB) colour value for each and every pixel in the image. There is no way for the system to fail recognizing the colour value respectively.

Then, the success rate of image binarization is fully depend on the image created by the grayscale function. This is because the grayscale function has set the pixel value to the range from only 0 to 255. The binarization process will scan through every pixel in the image and get its value. If the input image is not in grayscale format, then the pixel value might not be between 0 to 255. Hence, the system will do nothing about the pixel.

License plate detection has the success rate of 89%. The variable that drag down the success rate is the size of the license plate and the location of the character. Although the measurement of vehicle license plate is the standard in size in compliance with Malaysian government, there is still possible for some unexpected exceptions happen when the allocation of the characters is just slightly out range of the estimated width and length as shown in Figure 4.1.

Figure 4.1: Detected license plate draw with red rectangle

Figure 4.1 demonstrated the successful detected license plate's characters (dotted rectangle box) however last character is omitted. This is because the system found the license plate according to the white connected pixel until it reaches the estimated height and width.

Next, character segmentation is having the 97% success rate assume that the license plate is detected and cropped correctly. For the situation where there is a tiny region made up of black pixel but it is not belongs to any character on the cropped license plate, the system is performed a validation by considering the found region as a character only when the whole object is met with the minimum height. However, the system cannot differentiate the characters which are having the gap too close among each other as shown in Figure 4.2.

Figure 4.2: Segmented character that combines four characters

Suppose there are four characters that should be separated which are 'X', 'B', '7' and '2', but the system can only recognize them as one because the black pixel between them is not removed and they are connected. Only 3 tested images have faced this problem and 3 of the images is cropped from the square license plate.

Character recognition is the last step of the entire license plate recognition process which is fully depending on the segmented character. If the character is not segmented properly may result in incorrect character recognized. However, the system still able to find the matching score despite the segmented image is made up from how many characters. The success rate of character recognition is 98% where the 2 % is affected by factors including the image made up from two or more character and the different font type is used. In Malaysia, the vehicle can be registered with special plate numbers such as Figure 4.3(a) and (b).

The system is difficult to recognize the character in another font type especially in Italic format where minority of the vehicle is registered with the special words in Malaysia such as 'Putrajaya', 'BAMBee', 'Perodua' and so on. This is because the character template is predefined from the standard font type where the majority of the Malaysia registered vehicle own which is Calibri (bold).

DISCUSSION

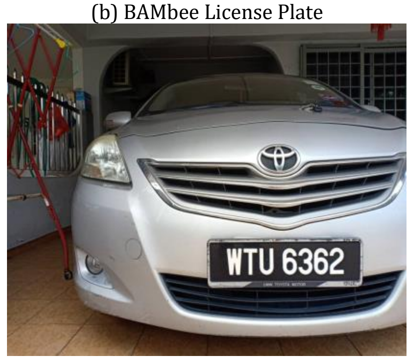
This solution worked well in an indoor car park but not in open area under the sun due to image captured under the sunshine condition is different comparatively. The consideration is the outdoor image can be brighter than the indoor image with less sunlight. If the image captured under the sun, the grayscale value can be much brighter and may affect the binarization process. The sample outcome demonstrated in Figure 4.3(c) and (d). Figure 4.3(d) is much brighter than Figure 4.3(c) as it was captured in outdoor. As a result, the entire car is becoming almost white at all after grayscale. As the binarization threshold value is set at 77, which is the most optimum value for the indoor environment.

(a)Putrajaya License Plate

(b) BAMbee License Plate

(c) Indoor Greyscale Image

(d) Outdoor Grayscale Image



(e) Right-aligned License Plate

(f) Top-left Angle

Figure 4.3: Sample Images

However, the threshold value of 77 is not suitable for the system operating at outdoor because the grayscale value should be brighter, around 135. The accuracy of the license plate detection can be affected as the threshold value is not fit.

In addition, the system is able to recognize the license plate regardless the location of license plate in the image. Typically, the license plate can be positioned either on the left, center, or right of the image. Figure 4.3(e) is the right-aligned license plate. Nevertheless, the license plate can be binarized and detected successfully despite its location in the image. This is achieved the image is scanned using a window sliding technique from left to right until it found the license plate.

However, the system is unable to recognize license plate that is having its angle twisted over the maximum as shown in Figure 4.3 (f). The failure of detecting the license plate is the current algorithm uses a search mechanism that finds the white connected pixel horizontally and vertically only. In case the captured image is either top left or top right angled, searching through the white connected pixel can be failed resulting the license plate cannot be detected.

CONCLUSION

In a nutshell, there are available algorithms as options for the LPR technology, ranging from the image acquisition to the character segmentation to the final optical character recognition. Each phase has its own different logic to achieve the desired result. They are closely connected as it is a linear process in order to achieve the result. In the other words, they are fully depending on each other. The most significant parts are the binarization and license plate detection, if the system cannot detect the license plate well, then the rest of the system can become useless.

This solution design provides the simplicity and also increased the performance comparing to the other similar systems. The process is simplified without excessive processing cycle such as noise reduction, gaussian filter, and edge detection since then the expected result can be achieved in an efficient manner. In this paper, the developed algorithm contributed an efficient character recognition over the license plate images.

One of the limitations of the system is its flexibility. The algorithm is developed by considering the scope in an indoor car park. As such, it is not suitable to be used in an outdoor environment as the captured image can be different. In addition, the current work demonstrated the algorithm of license plate detection and characters recognition can be simplified meantime the result is also achieved. From this, characters can be obtained on the predefined format i.e., the character-gap and also introduce several challenges on informal license plate such as characters are arranged too close among each other.

REFERENCES

1. Abdella, A.A.E.S, "Libyan Licenses Plate Recognition Using Template Matching Method," *Journal of Computer and Communications,* vol. 4, pp. 62-71, May. 2016.
2. Balaji, G.N. & Rajesh, D, "Smart Vehicle Number Plate Detection System for Different Countries Using an Improved Segmentation Method," *Imperial Journal of Interdisciplinary Research (IJIR)*, vol. 3, No. 6, pp. 263-268, 2017.
3. Koo, K. et al., "Character Segmentation and Recognition Algorithm of Text Region in Steel Images, " *Proceedings of the 8th WSEAS International Conference on Signal Processing, Robotics and Automation, Pohang, Korea,* 2009, pp 293-298.
4. Mayasari, N. & Siahaan, A.P.U., "Vehicle Plate Recognition using Template Matching, "*International Journal for Innovative Research in Multidisciplinary Field*, vol. 4, No. 10, pp. 259-263, Oct. 2018.
5. Mohammad, F., Anarase, J., Shingote, M. & Ghanwat, P., "Optical Character Recognition Implementation Using Pattern Matching," *International Journal of Computer Science and Information Technologies*, vol. 5, No. 2, pp. 2088-2090, 2014.
6. Patel, C., Shah, D. & Patel, A., "Automatic Number Plate Recognition System (ANPR): A Survey," *International Journal of Computer Applications,* vol. 69, No. 9, pp. 21-33, May. 2013.
7. S.Kranthi, K.Pranathi & A.Srisaila, "Automatic Number Plate Recognition," *International Journal of Advancements in Technology,* vol. 2, No. 3, pp. 408-422, Jul. 2011.
8. S.Vijayarani, D. & A.Sakila, M., "Template Matching Technique for Searching Words in Document Images," *International Journal on Cybernetics & Informatics (IJCI),* vol. 4, No. 6 , pp. 25-35, Dec. 2015.
9. Shrivakshan, G. & Chandrasekar, D., "A Comparison of various Edge Detection Techniques used in Image Processing, " *IJCSI International Journal of Computer Science Issues,* vol. 9, No. 5, pp. 269-276, Sep. 2012.
10. Soon, C.K., Lin, K.C., Jeng, C.Y. & Suandi, S.A., " Malaysian Car Number Plate Detection and Recognition System," *Australian Journal of Basic and Applied Sciences,* vol. 6, No. 3, pp. 49-59, Mac. 2012.
11. Swaroop, P. & Sharma, N., "An Overview of Various Template Matching Methodologies in Image Processing," *International Journal of Computer Applications,* vol. 153, No. 10, pp. 8-14, Nov 2016.
12. Tayo, A.O. & Gideon, A.D., "Vehicle License Plate Recognition Using Edge Detection and Nueral Netwrok," *International Journal of Science and Research (IJSR),* vol. 4, No. 4, pp. 3158-3161, Apr. 2015.
13. V.Karthikeyan, R.Sindhu, K.Anusha & D.S.Vijith., "Vehicle License Plate Character Segmentation – A Study," *International Journal of Computer and Electronics Research,* vol 2, No. 1, pp. 47-54, Feb. 2013.
14. V.Karthikeyan, V.J. Vijayalakshmi & P. Jeyakumar, "License Plate Segmentation Using Connected Component Analysis." *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. 4, No. 5, pp. 18-24, Jan-Feb. 2013.
15. X, A. D. et al., "License Plate Localization by Sobel Vertical Edge Detection Method," *International Journal of Emerging Technologies in Engineering Research (IJETER),* vol. 4, No. 6, pp. 48-53, Jun. 2016.
16. Zheng, L., He, X., Samali, B. & Yangc, L.T., "An algorithm for accuracy enhancement of license plate recognition, "*Journal of Computer and System Sciences,* vol. 79, No. 2, pp. 245-255, Mac. 2013.
17. Shrivastava, S., Jeyanthi, P.M. and Singh, S., 2020. Failure prediction of Indian Banks using SMOTE, Lasso regression, bagging and boosting. Cogent Economics & Finance, 8(1), p.1729569.
18. Anton Syroeshkin, Olga Levitskaya, Elena Uspenskaya, Tatiana Pleteneva, Daria Romaykina, Daria Ermakova. "Deuterium Depleted Water as an Adjuvant in Treatment of Cancer." *Systematic Reviews in Pharmacy* 10.1 (2019), 112-117. Print. doi:10.5530/srp.2019.1.19