Automatic Number Plate Recognition of Saudi License Car Plates

Rayana Antar
Department of Computer Science
Umm Al-Qura University
Jeddah, Saudi Arabia
s44286241@st.uqu.edu.sa

Shimaa Alghamdi
Department of Computer Science
Umm Al-Qura University
Makkah, Saudi Arabia
s44286664@st.uqu.edu.sa

Johara Alotaibi
Department of Computer Science
Umm Al-Qura University
Jeddah, Saudi Arabia
s44285136@st.uqu.edu.sa

Manal Alghamdi
Department of Computer Science
Umm Al-Qura University
Jeddah, Saudi Arabia
maalghamdi@uqu.edu.sa

Received: 30 December 2021 | Revised: 21 January 2022 | Accepted: 25 January 2022

Abstract-Automatic license plate recognition has become a significant tool as a result of the development of smart cities. During the experiment studied in the current paper, 50 images were used to detect Saudi car plates. After the preprocessing stage, the canny edge method to detect the car edges and different threshold techniques were used to reduce noise. Horizontal projection was applied in the segmentation process to split the plate. After that, a masking technique was utilized to locate and separate the region of interest in the image. OCR was applied to the processed images to read the characters and numbers in English and Arabic separately. Then, combining the English and Arabic text, after using the re-shaper for the Arabic letters. Finally, rendering of the results of text on images down the plate regions took place. The canny algorithm with projection technique with a proper preprocessing for images produces results with accuracy of 92.4% and 96% for Arabic and English language respectively.

Keywords-computer vision; edge detection; segmentation; OCR; license plates; recognition system

I. INTRODUCTION

Traffic regulation and traffic violations are major issues. The flow of traffic is regulated by a network of cameras placed across the streets. Since these cameras capture immediate images of cars, sophisticated software is required to recognize vehicles license plates [1]. In order to automatically record and recognize license plates, License Plate Recognition (LPR) or Automatic Number Plate Recognition (ANPR) are extensively employed in entrance/exits, parking lots, road traffic, security control of restricted areas, and traffic surveillance [2]. Many image processing applications have become more reliable and effective as a result of the usage of computer image processing. One of them is ANPR. Several recent studies detected vehicle's plates by applying different computer vision techniques. Authors in [3] proposed a new security system to trace stolen vehicles by capturing the plate numbers. They used canny edge detection to detect the plate and applied the Multi-Layer Perceptron Artificial Neural Network model. The result was displayed as text. Their proposed model achieved 97.8% accuracy in detecting multi-style Arabic characters. Authors in [4] proposed an identification system based on vehicle license and number plate recognition. The system's LNPR software employs a series of image processing algorithms to recognize number plates and identifies the vehicle from a database. The median filter was employed to minimize the image's visual noise. Authors in [5] provided a Sobel edge detection technique and morphological operation for ANPR. The bounding box method was utilized to segment the numbers and letters on the plate. To distinguish numbers and characters after segmentation, a template matching technique was employed. Authors in [6] introduced the ISeeCarRecognizer, an automated recognition system for reading Vietnamese registration numbers using boundary line-based technique integrating the Hough transform and the Contour algorithm in the VLP detection module. For segmentation, horizontal and vertical projections were used to separate plate numbers. Finally, an Optical Character Recognition (OCR) module based on a Hidden Markov Model detects the plate number. Authors in [7] cropped 9021 License Plate (LP) images of 5 different countries. Image segmentation was used with a CNN to detect the language and the country of the LP with 99.5% accuracy. The Secondary Positioning (SP) model was utilized in [8]. A plate number localization technique was presented and evaluated using locally obtained data. The position of a plate number was determined by scanning the red light areas in HSV color space, and the plate number was localized by determining the plate number's vertical edge. A correction coefficient was generated between the templates and the testing pictures and template matching was used to recognize individual characters with a precision of about 75% and 70%.

Corresponding author: Manal Alghamdi

Italian-style automobile registration plate recognition was conducted in [9]. To determine the plate and character locations, images were acquired from a toll gate camera and were preprocessed via a quick and robust 1-D DFT method. A multilayer neural network using the recently presented BRLS learning method was used to classify characters. Authors in [10] designed a smart system for recognizing license plates, by applying canny edge detection to identify the LP location. The images were collected locally with a camera and the output was segmented into numbers and letters by the binary K-means algorithm, with 90% accuracy for English LP recognition. Authors in [11] proposed an automatic Iraqi car LP identification model by utilizing OCR. The model can extract the plates' features by dividing the numbers and words into sub-images and achieved 86.6% accuracy. Authors in [12] proposed an improved automatic LP recognition system, in which 500 Jordanian vehicle images were processed. In the detection stage, histogram vertical-edge analysis and potential region size estimation were performed. During the segmentation, the mismatched license plate images were de-rotated and the feature extraction results of the plate's letters reached 91.5% accuracy.

The purpose of this paper is to collect a dataset of cars in the Kingdom of Saudi Arabia (KSA) and to perform recognition of Saudi car plates with the use of edge detection, segmentation, and contouring techniques. In addition, a preprocessing technique of the collected images is utilized. After that, OCR is utilized to extract letters and numbers from the processed images. The proposed approach aims to detect different car plate shapes and recognize the plate in both Arabic and English texts.

II. METHODOLOGY

The basic tasks of the LPR system are to find the LP and to recognize the LP characters. Number plate identification algorithms are classified into many categories depending on the different methodologies they use [13]. The following variables should be considered while detecting vehicle number plates:

- The size of the car plate may vary in size from one image to another.
- The location of the plate on the vehicle.
- The background color of a plate might vary depending on the vehicle.
- A plate might have a screw or motto or symbol, which could be regarded as a character.

The flowchart of the proposed system is shown in Figure 1. The following methodologies were used in LPR.

A. License Plate Extraction

The image processing stage is important for the enhancement of images [14]. Resizing, scaling, and rotation were performed on the images. The input images were converted to grayscale, the RGB values were transformed from 24-bit onto 8-bit gray level values by adding or removing the alpha channel. The brightness of a grayscale picture pixel value varies from 0 to 255 [14]. Moreover, to create a smoother image, the interpolation method was used in image processing. In picture interpolation, the rule is to utilize a source image as a reference to create a new or interpolated/scaled image [15]. The size of the produced image is determined by the specified interpolation proportion. When we execute the interpolation, we produce empty areas in the original image and fill them with the proper pixel values. As a result, the interpolation processes provide varied outcomes. We used the nearest neighbor approach to replace empty areas with the next nearby image pixels.

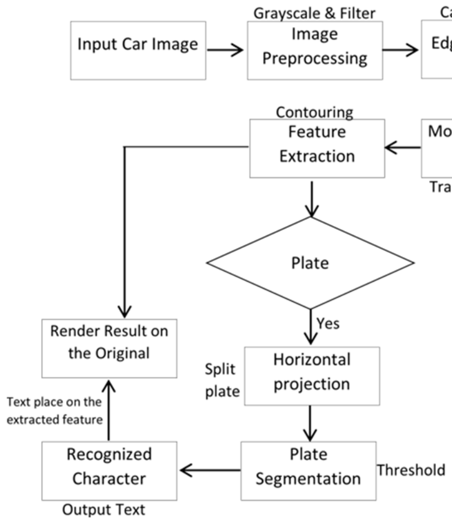

Fig. 1. Recognition system flowchart.

Filters are primarily used to reduce the high frequencies within a picture [16]. The bilateral filter technique is used to preserve the edges of images. The basic idea is to include a photometric weight into a regular Gaussian filter. This weight has the effect of canceling the spatial connections between pixels with significant intensity differences. Suppose a photometric weight w_p is incorporated as a factor of the spatial weight w_i. The bilateral filter produces the picture F from the source image E as follows:

$$F(x) = \frac{\sum_{t\in I_m} w_i(||t||)w_p(E(x)-E(x+t))E(x+t)}{\sum_{t\in I_m} W_i(||t||)w_P(E(x)-E(x+t))} \quad (1)$$

where I_m represents a square window $[-m,m] \times [-m,m]$, w_i is a decreasing symmetrical function of the distance k_t from the center of I_m, and w_p is decreasing the intensity of the function.

The edge information of an image explains the target boundaries, the location inside the region of interest [17], and other significant details [18]. First derivatives, such as Sobel, are used to detect the gradient by calculating the minimum and maximum in the first derivative input images [19], which are very sharp edges [20]. The canny edge, which is utilized in the experiment, is a multi-stage process used in edge detection. It has minimum error rate and reduces noise [22]. The algorithm involves the following steps:

- Converting the original image to grayscale.
- Applying Gaussian blurring to remove high-frequency noise that may be detected as false edges.

- Compute the gradients along the x and y axes (G_{x1},G_y) independently.
- The final gradient over the whole image can be computed using the follow equation [20]:

$$\sqrt{G_x^2 + G_y^2} \quad (2)$$

where the G_x and G_y are the mask values of the operator [21] as shown in Table I.

TABLE I. MAX VALUES OF G_X AND G_Y

Mask Values of G_x	Mask Values of G_y
$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

- Non-maximum suppression is conducted, where only pixels that constitute a local maximum in their neighborhood are considered, and the other pixels are set to zero. This results in a binarized image.
- Hysteresis thresholding follows, where the two threshold values are high and low threshold. All pixels with intensity values higher than the high threshold are picked as strong edges, and all intensity values lower than the low threshold are discarded. Pixels with values between the two thresholds are only picked if they are connected to a strong edge.

In practice, the picture's edges represent a small portion of the image information. In photos with less edge information, the mean gradient magnitude and standard deviation of most of the pixels are placed in a limited range [23]. Because the gradient magnitude distribution of those non-edge pixels is concentrated, a good threshold can help pick edge pixels out. Equation (3) is the threshold selection approach for photos with little edge information [24]:

$$T_I = T_h /2 \quad (3)$$

where T_l and T_h represent the low and the high threshold respectively [23].

After that, contours were used to detect lines and polygons. We utilized a contour with 4 points like the plates' shape, to find the shape from the edge information.

B. License Plate Recognition

After extracting the plate, the projection profile is generated independently for each axis. The projection profile approach is primarily used for text object segmentation within the text [25]. The horizontal projection feature is the picture's projection profile along the horizontal axis. For each row, the horizontal projection profile is determined as the sum of all column pixel values within the row [25]. To turn printed words of the preprocessed images into editable text, the OCR technology is used. Image preprocessing and segmentation processes can have an impact on OCR accuracy [25]. Character recognition necessitates the matching of the resulting binary format with the existing template. A matching track-sector matrix must be produced, defining the number of pixels in each region and the matrix's center [27], by calculating the radius via distance equations to identify the pixel with the greatest distance from the center as in (4):

$$D = \sqrt{\left((y_2 - y_1{}^2) + (x_2 - x_1{}^2)\right)} \quad (4)$$

The size of each track matrix is identified via calculating number of 1's in each intersection of sector and track [27].

C. Evaluation Mertrics

The evaluation metrics that were employed in this study to evaluate the performance are precision, recall, F1-score, and Mean Average Precision (MAP) to measure the detector performance for English and Arabic text separately. When the costs of False Positives (FP) are substantial, Precision is a suitable metric to use [29]. Precision is defined as:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

When there is a large cost related to False Negatives (FN), the Recall measure is used to identify the optimum model, and it's calculated as:

$$Recall = \frac{TP}{Total\ Actual\ Positives} \quad (6)$$

For instance, a correct plate number (Actual Positive) passes the test but is predicted to be incorrect (i.e. FN). If the car was stolen, the cost of FN will be exceedingly significant.

F1 is a function of Precision and Recall and is calculated as in (7). Whenever we want to strike a balance among Precision and Recall, the F1-score achieve this balance.

$$F1 = 2 \text{ x } \frac{Precision\ x\ Recall}{Precision+Recall} \quad (7)$$

Accuracy is the simplest basic performance metric, which is just a ratio of accurately outcome expectations to the total number of samples, and is calculated by:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

III. EXPERIMENTS

This section contains a summary of the obtained data as well as information about Saudi traffic laws and the automobile LP system. Furthermore, the framework configuration is described in depth.

D. Dataset

The dataset contains 50 images of various parked cars. An iPhone 12 Pro Max camera was used to capture the images, which has a resolution of 1284×2778 pixels. The pictures were taken from different positions (front and back), different angles, various time intervals, and different light conditions. In addition, these images were captured inside/outside the parking as shown in Figure 2, representing a sample of the collected dataset. The Saudi car plates have two sizes (w×h), namely 30cm×16cm and 53cm×14cm. Also, the LPs combine Arabic and English characters and numbers. The top part of the plate contains the Arabic language (the top right has 3 Arabic characters and the top left has 4 Arabic numbers), whereas the bottom part contains 3 English characters at the right and 4 numbers at the left as shown in Figure 3.

Fig. 2. Dataset images that were collected during (a) the morning, (b) the afternoon, and (c) the night periods, and (d) various tilt angles.

Additionally, Saudi Arabia doesn't support all Arabic letters for security purposes. For example, there are two similar letters, غ and ع. Some criminals may take advantage of these and just remove the color from the dot and change the plate. Another reason is the need to match the number of Arabic letters with the number of English letters. Figure 4 depicts the Arabic and English characters that are used in Saudi plates.

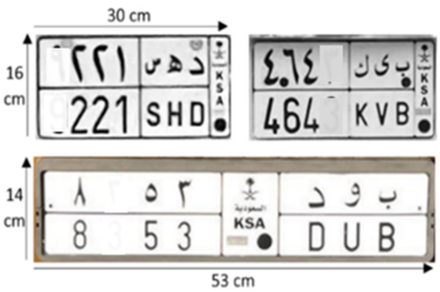

Fig. 3. Some Saudi license plates.

Arabic Letter	English Letter
ا	A
ب	B
ح	J
د	D
ر	R
س	S
ص	X
ط	T
ع	E
ق	G
ك	K
ل	L
م	Z
ن	N
هـ	H
و	U
ى	V

Fig. 4. Approved letters in the Saudi license plates.

E. Experimental Setup

The experiment was conducted on the Jupyter program [28]. Jupyter Notebook is an open-source software that allows collaborative data science, used to organize, and execute models, which were written in Python. The specification of the device on which these programs are installed is Windows 10 Operating System on an Intel Core (TM) i7-1065G7 CPU 1.50GHz with 16GB RAM to test the algorithm's effectiveness.

F. Plate Detection and Localization

To detect the plate location, proper scaling of images that helps reduce the number of pixels in the photos is performed. Figure 5 depicts different interpolation methods for resizing. The Interpolation Nearest method was chosen.

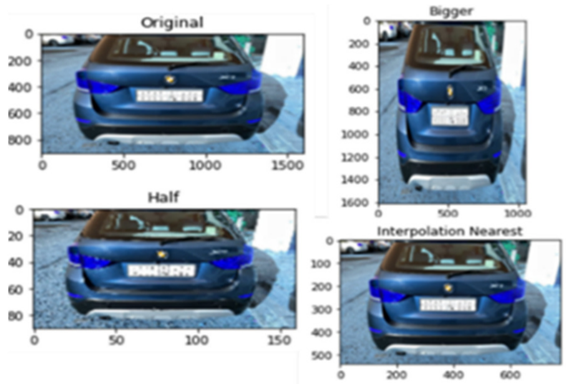

Fig. 5. Different interpolation techniques.

After that, the images were convolved using Gaussian blur to remove the high frequencies with a kernel size of 5×5. A bilateral filter technique transforms a filter into a bilateral picture. The intensity, noise and smoothing values were set to 60, 80, and 30 respectively, and the output is shown in Figure 6. The parameters for intensity, noise and smoothing should be set very high because when using a small value, the other details appear.

Fig. 6. The output of the bilateral filter.

In canny detection, the parameter was set very low in order to not detect other areas like floor and car details, so 20 was chosen for all images as shown in Figure 7. We can observe that other details are shown (Figure 8) if we set the parameter too high. Morphological Gradient was performed to find the outline of an object as shown in Figure 9. Moreover, the contours were used to detect within lines and polygons. We used contours with 4 points like the plates' shape. In addition, we utilized masking to separate the remaining edges to transfer the pattern. The plate should generate a blank mask. The final image is presented in Figure 10. Some of the images in the contouring process could not detect the complete plate, but only a part of it due to lighting, angles and texturing conditions. Figure 11 shows the result of the detected plate.

Fig. 7. Canny detector with low parameter value.

Fig. 8. Canny detector with high parameter value.

Fig. 9. The output of the morphological gradient.

Fig. 10. Detecting the care plate.

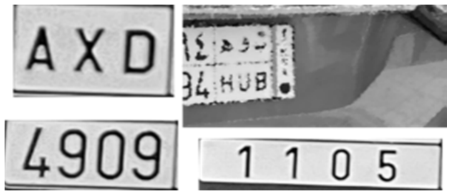

Fig. 11. Displaced detection cases.

G. Plate Number Recognition

Before placing the images into the OCR model as inputs, the plate needs to be divided into rows, so horizontal projection is performed. Figure 12 shows the result of the projection on the plate.

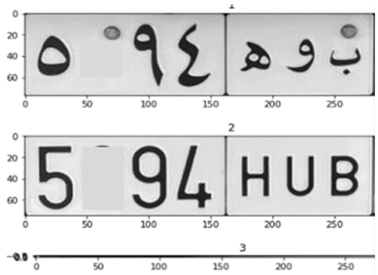

Fig. 12. Horizontal projection.

Figure 13 represents the supervised segmentation by thresholding with different threshold values over the plate. The threshold value 3 was chosen since it does not show the nails. After the processing and noise reducing, the images are ready to be inserted into the OCR reader without training. We utilized the engine mode (–oem) of the OCR model. For the Arabic language, a re-shaper is needed to show the letter correctly when rendering text on the images. The final render result is shown in Figure 14.

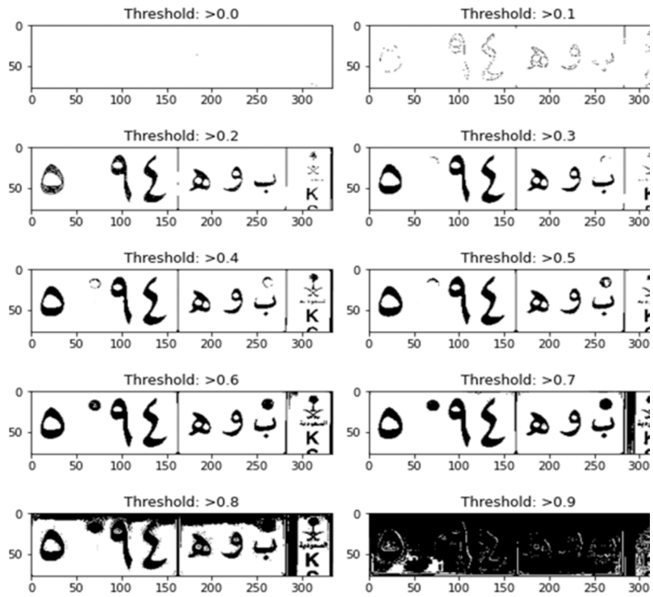

Fig. 13. Various threshold values on processed plates.

Fig. 14. Rendering text result on the original image.

H. Comparison with Related Works

According to the conducted review, some researchers used techniques like Artificial Neural Networks (ANN) and Machine Learning (ML) for the purpose of reading the characters and numbers in various languages. The findings are shown in a detailed comparison in Table II, whereas the authors in [3] achieved a high accuracy ratio for Arabic plates with the use of the Multi-Layer Perceptron (MLP) approach.

TABLE II. RESULT COMPARISON

Reference	Techniques used	Dataset	Result
[3]	Canny edge detection with MLP	Arabic	97.8%
[8]	ANN	English	75%
[10]	Canny edge. K-means algorithm	English	90%
[9]	Multilayer ANN	English	90%
[17]	YOLOv2 detector with ResNet5 to detect LPs only	5 languages	99.5%

Table III depicts the comparison of the results regarding plates in Arabic. Authors in [12] achieved 91.5% accuracy but our model obtained 92.4% on our collected dataset. Regarding LPs in English, Table IV compares the results of similar works with our proposed method. Our proposed method (consisting of the canny method, along with a projection approach and with the suitable image processing techniques that we applied), outperforms the others by achieving 96% accuracy. Figures 15 and 16 exhibit the average Precision result for LPs in English and Arabic respectively.

TABLE III. ARABIC LP RECOGNITION RESULT COMPARISON

Reference	Techniques used	Dataset	Result
[4]	OCR - by comparing with the records on a database	Iranian	-
[11]	OCR	Arabic	86.6%
[12]	Canny edge detection & Sobel edge detection	Arabic	91.5%.
Current study	Canny edge detection with horizontal projection - OCR	Arabic	92.4%

TABLE IV. ENGLISH LP RECOGNITION RESULT COMPARISON

Reference	Techniques used	Dataset	Result
[5]	Sobel edge. Morphological operation	English	-
[6]	Boundary line-based technique. Contour algorithm. OCR	English	92.85%
Current study	Canny edge detection with horizontal projection - OCR	English	96.0%

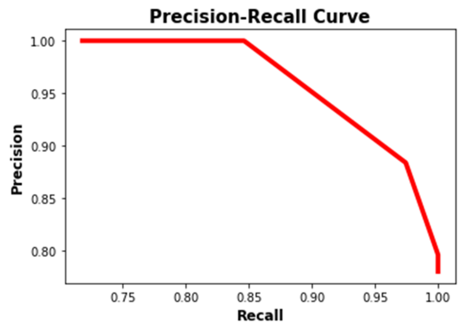

Fig. 15. Average Precision for English LPs.

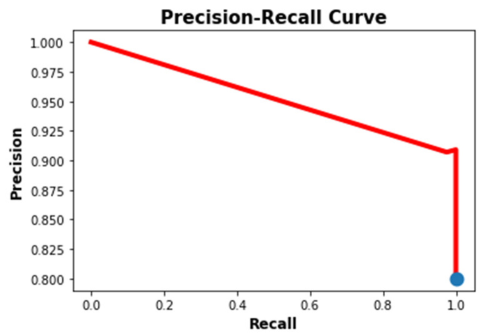

Fig. 16. Average Precision for Arabic LPs.

IV. CONCLUSION

In the current paper, after the preprocessing stages, the canny edge approach was used to detect the plate edges, and multiple threshold strategies were employed to minimize the image noise. The plate was divided into rows using vertical projection during the segmentation process. After that, the masking technique was used to find and divide the image's regions of interest. The letters and numerals in English and Arabic are read individually using OCR on the processed pictures. Then, the re-shaper for the Arabic characters was used, combining English and Arabic text. Finally, the effects of the text on images were rendered along with the plate areas. The canny method, along with a projection approach and suitable picture preparation, results in an accuracy of 92.4% for Arabic and 96% for English texts on LPs.

Future work will concern training the model in various conditions to increase the obtained accuracy results and increasing the dataset size. Also, the model could be developed to recognize all types of cars' plates by using different digital cameras to get good results. In addition, training the model with natural language processing on the Arabic letters to increase the speed of operation in real-time applications will be considered.

ACKNOWLEDGMENT

We wish to thank the car owners who allowed us to take photos of their cars, the completion of our study would not have been possible without their support.

REFERENCES

[1] N. O. Yaseen, S. Ganim Saeed Al-Ali, and A. Sengur, "Development of New Anpr Dataset for Automatic Number Plate Detection and Recognition in North of Iraq," in *1st International Informatics and Software Engineering Conference*, Ankara, Turkey, Nov. 2019, pp. 1–6, https://doi.org/10.1109/UBMYK48245.2019.8965512.

[2] N. Rana and P. Dahiya, "Localization Techniques in ANPR Systems: A-State-of-Art," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 5, pp. 682–686, Feb. 2017, https://doi.org/10.23956/ijarcsse/SV7I5/0338.

[3] B. A. Hussain and M. S. Hathal, "Developing Arabic License Plate Recognition System Using Artificial Neural Network and Canny Edge Detection," *Baghdad Science Journal*, vol. 17, no. 3, pp. 909–915, 2020, https://doi.org/10.21123/bsj.2020.17.3.0909.

[4] H. Saghaei, "Proposal for Automatic License and Number Plate Recognition System for Vehicle Identification," *Nuclear Physics B*, vol. 614, no. 3, pp. 467–493, Nov. 2001, https://doi.org/10.1016/S0550-3213(01)00405-9.

[5] F. Patel, J. Solanki, V. Rajguru, and A. Saxena, "Recognition of Vehicle Number Plate Using Image Processing Technique," *Control and Systems Engineering*, vol. 2, no. 1, pp. 1–7, Sep. 2018, https://doi.org/10.18063/cse.v2i1.405.

[6] T. Duan, D. Tran, P. Tran, and N. Hoang, "Building an Automatic Vehicle License-Plate Recognition System," in *International E-Conference on Computer Science*, Can Tho, Vietnam, Feb. 2005, pp. 59–63.

[7] M. Salemdeeb and S. Erturk, "Multi-national and Multi-language License Plate Detection using Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5979–5985, Aug. 2020, https://doi.org/10.48084/etasr.3573.

[8] J. Wang, B. Bacic, and W. Q. Yan, "An effective method for plate number recognition," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1679–1692, Jan. 2018, https://doi.org/10.1007/s11042-017-4356-z.

[9] R. Parisi, E. D. Di Claudio, G. Lucarelli, and G. Orlandi, "Car plate recognition by neural networks and image processing," in *International Symposium on Circuits and Systems*, Monterey, CA, USA, Jun. 1998, vol. 3, pp. 195–198 vol.3, https://doi.org/10.1109/ISCAS.1998.703970.

[10] J. Han and H. Y. Bo, "Analysis and Design of intelligent license Plate recognition system," in *6th International Conference on Intelligent Computing and Signal Processing*, Xi'an, China, Apr. 2021, pp. 1079–1082, https://doi.org/10.1109/ICSP51882.2021.9408766.

[11] S. S. Omran and J. A. Jarallah, "Iraqi car license plate recognition using OCR," in *Annual Conference on New Trends in Information &* Communications *Technology Applications*, Baghdad, Iraq, Mar. 2017, pp. 298–303, https://doi.org/10.1109/NTICT.2017.7976127.

[12] M. Al-Yaman, H. Alhaj Mustafa, S. Hassanain, A. Abd AlRaheem, A. Alsharkawi, and M. Al-Taee, "Improved Automatic License Plate Recognition in Jordan Based on Ceiling Analysis," *Applied Sciences*, vol. 11, no. 22, Jan. 2021, Art. no. 10614, https://doi.org/10.3390/app112210614.

[13] Lubna, N. Mufti, and S. A. A. Shah, "Automatic Number Plate Recognition:A Detailed Survey of Relevant Algorithms," *Sensors*, vol. 21, no. 9, Jan. 2021, Art. no. 3028, https://doi.org/10.3390/s21093028.

[14] C. Saravanan, "Color Image to Grayscale Image Conversion," in *Second International Conference on Computer Engineering and Applications*, Bali, Indonesia, Mar. 2010, vol. 2, pp. 196–199, https://doi.org/10.1109/ICCEA.2010.192.

[15] O. Rukundo and H. Cao, "Nearest Neighbor Value Interpolation," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 4, pp. 1–6, 2012, https://doi.org/10.14569/IJACSA.2012.030405.

[16] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Bombay, India, Jan. 1998, pp. 839–846, https://doi.org/10.1109/ICCV.1998.710815.

[17] L. Xuan and Z. Hong, "An improved canny edge detection algorithm," in *8th IEEE International Conference on Software Engineering and Service Science*, Beijing, China, Nov. 2017, pp. 275–278, https://doi.org/10.1109/ICSESS.2017.8342913.

[18] O. Vincent and O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection," in *Informing Science + IT Education Conference*, Macon, United States, Jun. 2009, vol. 9, pp. 97–107, https://doi.org/10.28945/3351.

[19] S. Israni and S. Jain, "Edge detection of license plate using Sobel operator," in *International Conference on Electrical, Electronics, and Optimization Techniques*, Chennai, India, Mar. 2016, pp. 3561–3563, https://doi.org/10.1109/ICEEOT.2016.7755367.

[20] N. Basheer, A. Aaref, and D. Ayyed, "Digital Image Sobel Edge Detection Using FPGA," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 7, pp. 183–190, Apr. 2015.

[21] H. Shimpi, M. Dhage, A. S. Pawar, and N. Gaikwad, "Implementation of Edge Detection Algorithm Using FPGA," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 4, no. 4, pp. 2193–2197, https://doi.org/10.15662/ijareeie.2015.0404080.

[22] R. Dhar, R. Gupta, and K. L. Baishnab, "An analysis of Canny and Laplacian of Gaussian image filters in regard to evaluating retinal image," in *International Conference on Green Computing Communication and Electrical Engineering*, Coimbatore, India, Mar. 2014, pp. 1–6, https://doi.org/10.1109/ICGCCEE.2014.6922270.

[23] S. Gupta and N. Mohan, "Color Channel Characteristics (CCC) for Efficient Digital Image Forensics," *Engineering, Technology & Applied Science Research*, vol. 8, no. 1, pp. 2555–2561, Feb. 2018, https://doi.org/10.48084/etasr.1744.

[24] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved Canny edge detection algorithm," in *IEEE International Conference on Mechatronics and Automation*, Tianjin, China, Aug. 2014, pp. 577–582, https://doi.org/10.1109/ICMA.2014.6885761.

[25] M. K. Mahto, K. Bhatia, and R. K. Sharma, "Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition," in *International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, Mar. 2015, pp. 59–65, https://doi.org/10.1109/ICACEA.2015.7164646.

[26] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, vol. 55, no. 10, pp. 50–56, Jul. 2012, https://doi.org/10.5120/8794-2784.

[27] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical Character Recognition Implementation Using Pattern Matching," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2088–2090, 2014.

[28] T. Kluyver *et al.*, "Jupyter Notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Scmidt, Eds. Amsterdam, Netherlands: IOS Press, 2016, pp. 87–90.

[29] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, "Analysis of Text Feature Extractors using Deep Learning on Fake News," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, https://doi.org/10.48084/etasr.4069.