

# Accurate license plate recognition system for different styles of Iraqi license plates

Sukaina Sh Altyar, Samera Shams Hussein, Lubab Ahmed Tawfeeq

Department of Computer Science, College of Education for Pure Science, University of Baghdad, Baghdad, Iraq

## Article Info

### Article history:

Received Jun 1, 2022

Revised Sep 21, 2022

Accepted Nov 5, 2022

### Keywords:

ALPR

Arabic license plate

Automatic license plate recognition

Pattern recognition

Plate recognition

## ABSTRACT

Automatic license plate recognition (ALPR) used for many applications especially in security applications, including border control. However, more accurate and language-independent techniques are still needed. This work provides a new approach to identifying Arabic license plates in different formats, colors, and even including English characters. Numbers, characters, and layouts with either 1-line or 2-line layouts are presented. For the test, we intend to use Iraqi license plates as there is a wide range of license plate styles written in Arabic, Kurdish, and English/Arabic languages, each different in style and color. This variety makes it difficult for recent traditional license plate recognition systems and algorithms to recognize all these license plate types using the same algorithm. In this work, a new method has been proposed to efficiently recognize all these types of license plates. This has been done by utilizing a series of algorithms for preprocessing and recognition with new identification strategies. The results show that the system recognized license plate numbers with higher accuracy, reaching up to 97.85%. However, the method field to detect license plates when there are some high deformations in plate numbers or when they are partially covered with mud, which makes it difficult to distinguish numbers.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Sukaina Sh Altyar

Department of Computer Science, College of Education for Pure Science, University of Baghdad  
Baghdad, Iraq

Email: Sukaina.s.m@ihcoedu.uobaghdad.edu.iq

## 1. INTRODUCTION

Since the number of vehicles is generally increasing every year, the process of controlling and tracking those vehicles represents a significant human challenge, and the automatic recognition of those vehicles can help solve this problem and save substantial effort and time [1], [2]. An automatic license plate recognition (ALPR) system is a form of automatic vehicle identification. ALPR has been designed to recognize the characters and numbers on vehicle plates. The ALPR process involves capturing a plate image from a vehicle and then applying several processes to recognize the characters and numbers on the plate. Typically, the process includes detection, image processing, and pattern recognition [3], [4]. In general, several countries use license plates that have English characters, and the numbers and letters are separated. However, some countries have their own license plate recognition style and some use their language (characters) in [5]. Since license plates are different and specific for every country, there is a wide range of algorithms that can be used depending on the license plate design [6]. In Iraq, there are two main license plate styles [7], [8], which are the old style in Figure 1(a) and the newest style in Figure 1(b), and both are still used. The old-style contains Eastern Arabic numerals and Arabic characters, and the newest style contains Arabic numerals, Eastern Arabic numerals, Arabic characters, and English characters. In addition to numbers and characters, there are colors that refer to the vehicle type. In the old style, the color filled the

entire license plate, while in the newer style, the color is on the left side of the license plate. More details of the Iraqi license plate styles are described in Table 1.

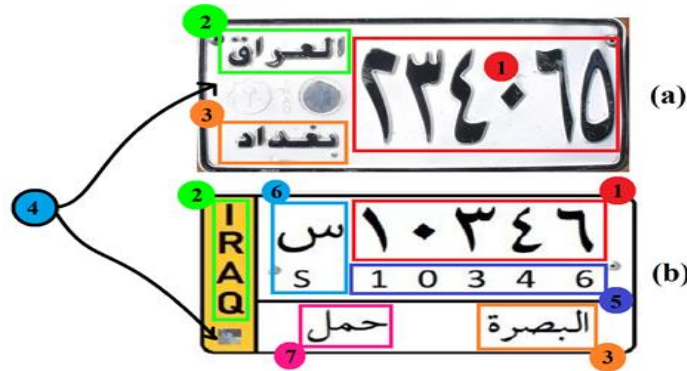


Figure 1. Iraqi license plates for (a) old style and (b) new style. The old style has four parts, while the newest style has seven parts

Table 1. Iraqi license plates: styles and descriptions [9]

Type	Color	Vehicle type	License plate shape	
Private	White	Any personal use cars, such as SUVs, sedans, 4-wheel-drive vehicles, and vans	Old style	
			New style	
Public	Red	Any Public use vehicles, such as buses, mini-buses, and taxis	Old style	
			New style	
Trade	Yellow	Any vehicles used for trade, such as trucks and heavy trucks	Old style	
			New style	
Governmental	Blue	All vehicle types	Old style	
			New style	
Temporary customs entry	Orange	All vehicle types	New style	
Agriculture	Green	Agricultural vehicles such as tractors and harvesters	New style	
Temporary permit		All vehicle types	Old style	

Explanation of the number from Figure 1:

1. License plate number, in Eastern Arabic numerals.
2. Iraq's country label is in Arabic characters in the old style but in English characters in the newest style.
3. Governorate, in Arabic characters.
4. plate color, which represents the car type (private: white, trade: yellow, public: red, and governmental: blue).
5. License plate number, in English numerals.
6. One English and one Arabic character.
7. Car types are written using Arabic characters: private, public (e.g., taxi or bus), trade (e.g., truck), governmental (e.g., ministry vehicles), and diplomatic.

More details of the Iraqi license plate styles are described in Table 1. As shown from Table 1, there is a wide range of Iraqi license plate styles, making recognition difficult [10], [11]. However, all Iraqi license plates use Eastern Arabic numerals, and thus, in this work, we focused on recognizing four aspects: the car type has one digit that is an English character, two Arabic characters representing the governorate and one color that represents the car type. All other details were neglected.

## 2. RELATED WORKS

Many license plate recognition systems have been developed in recent years. Li *et al.* [12] propose a deep neural network that can find the location of license plates and read the letters in a single forward pass. End-to-end training can be done on the whole network. Their model includes a variety of convolutional layers to obtain discriminating features for license plates, a fully connected network designed specifically for automobile license plates, a pooling layer for the region of interest (RoI); multi-layer perceptrons for plate detection and bounding box regression, and recurrent neural networks (RNNs) with connectionist temporal classification (CTC) for plate recognition. Plate detection and recognition can be carried out concurrently using this design, which only requires one network and a single evaluation of the input image in its forward direction. In addition, the entire network is trained end-to-end, meaning that both the localization loss and the recognition loss are jointly optimized. As a result, the performance of the network has been improved.

In the sections that follow, we will provide a comprehensive description of each individual component. Existing methods look at license plate detection and recognition as two separate tasks and solve them one at a time. For testing performance, three sets of images taken from different scenes and under different conditions are used. A lot of tests have shown that our proposed method works and is efficient. The results show that the model achieved 96.13% in end-to-end performance and 98.15% in detection-only performance when using the carflag-large dataset with a processing time of about 300 ms. When used with the Aolp dataset, the model achieved 95.29% accuracy in end-to-end performance and 99.56% accuracy in detection-only performance with a processing time of about 400 ms. Xie *et al.* [13] present a novel approach for high-accuracy real-time vehicle license plate detection that is based on convolutional neural networks (CNN) for the detection of car license plates in multiple directions. The rotational issues that arise in real-time scenarios can be deftly handled by our proposed method thanks to its accurate rotation angle prediction and its lightning-fast intersection-over-union evaluation strategy. A number of tests have been carried out to demonstrate that the method is superior to other existing methods. The results show the model achieved 98.32% when using the UCSD dataset and 97.38% when using the phenylketonuria (PKU) dataset, and the processing time is (530+140) ms when using the central processing unit (CPU) and (5+2.7) when using the graphics processing units (GPU). These results show that the state-of-the-art model achieves more accurate results at a lower cost to the computational system.

Srinu *et al.* [14] proposed the concept a model-based template match. Adaptive thresholding is used to perform the initial step of the image's preprocessing. On the threshold image, the connected-component analysis with the use of bounding boxes will be carried out. Statistical and logical operations are applied in order to extract the candidate region for the maximum magnitude row of an image based on the pixel value, which is used to highlight the desired plate details. Following the acquisition of the candidate region, the recognition of the license plate number was carried out utilizing template matching. The results were tested for 1,200 vehicles, and the testing results show that the model achieved 96.6% accuracy with good prediction. The results also show that the proposed method has a shorter amount of time required for recognition when compared to the other recent methods, which is about 35 ms. A new method for determining the location of a car's license plate based on the new model you only look once-L (YOLO-L) and plate pre-identification has been proposed by Min *et al.* [15]. The new model is an improvement in two different ways, both of which help to pinpoint the location of the license plate. First, a k-means++ clustering algorithm is used to determine the optimal range for the number of plate candidate boxes and their sizes. Second, the YOLOv2 model's structure and depth are both altered as a result of this. The algorithm used for

plate pre-identification is able to effectively differentiate license plates from other objects that are visually similar. The results of the experiments show that the authors' proposed method not only outperforms the methods that are already in use by achieving a precision of 98.86% and a recall of 99.86%, but it also has a high level of efficiency when applied in real time.

Alhamza and Alaythawy [16] proposed a model to recognize license plate cars with numbers or characters in the plate in situations in which it was hard to detect the same object in different environments. In addition to this, it provides license plate recognition, which is comprised of three primary stages: the first is preprocessing, which includes detecting license plates; the second is segmentation; and the third is character recognition. The first step is to take a picture with the camera, and then to perform some preliminary processing on it. Check the image for a matching license plate using the license plate detection feature, then crop the image to include only the appropriate plate. The numbers were first split up in order to perform the segmentation separately. The last step is called number recognition, and it is accomplished with the help of an algorithm called k-nearest neighbors (KNN). This is one of the fundamental machine learning algorithms, and it is utilized to match numbers with training data in order to provide an accurate prediction. Python 3.5 and the OpenCV library were utilized in the development of the system. When tested with a set of 50 images, the system achieved a performance level of 90% accuracy.

Zandi and Rajabi [17] proposed a framework of deep CNNs for reading Iranian license plates. The first CNN is the YOLOv3 network, which finds the Iranian license plate in the input image. The second CNN is a faster region-based CNN (R-CNN), which recognizes and sorts the characters in the found license plate. In this paper, we also made a set of images of Iranian license plates that were not in good shape. The YOLOv3 network got 99.6% mAP, 98.26% recall, and 98.8% accuracy, and the average time it takes to find something is only 23 ms. Also, the faster R-CNN network was trained and tested on the set of data that was created, and it got 98.97% recall, 99.9% precision, and 98.8% accuracy. The proposed system can read the license plate even when there are other things on the plate that make it hard to read. When this system is compared to other Iranian license plate recognition systems, it is faster and more accurate.

### 3. PROPOSED AUTOMATIC LICENSE PLATE RECOGNITION METHOD

The typical ALPR system consists of two parts: hardware and software [18]. The hardware components are used to detect vehicles and process the overall operations. This part requires some hardware components. The most important and main hardware component is the camera, which is used to capture images of vehicle license plates. There is a wide range of cameras that can be used. However, a camera with a high resolution and a fast response time is recommended. The second most important piece of hardware is the processing unit (in this work we utilized the Raspberry Pi official camera V2-8 megapixel). The processing unit can be a computer unit, a single board computer, and a microcontroller (in this work we utilized the Raspberry Pi v3b [19]). Another hardware component that can be used is sensors. These can be used in specific applications, such as on borders and in security zones that vehicles may not pass through all the time. These sensors can be motion detection sensors or radar [20]. The second part is the software, which is an important portion of this system in this model. It includes a series of image processing techniques. Typically, the software development process consists of three stages. The first stage is image acquisition and vehicle detection. At this stage, the software gathers streaming video from a camera and then recognizes and captures the vehicle. The second stage detects and extracts the plate region from a vehicle. The third stage is character and word recognition. This stage is done after the license plate region is extracted. The individual characters and numbers on the license plate must be extracted and recognized. The most important process in this stage is to detect and recognize the numbers, characters, and words on the license plate. The proposed ALPR system is designed using the MATLAB program. The system strategy is based on a series of processes to recognize Arabic (Iraq) license plates, which are as follows:

#### 3.1. Image acquisition

In this stage, the system will get streaming video from the camera. The algorithm is as follows: Step 1: read all the frames in the video; Step 2: size a figure based on the video's width and height; Step 3: find the object using the depth camera; Step 4: capture 1 frame every time the depth video is triggered (the camera is 25 fps); Step 5: trigger the depth video to capture the image; Step 6: display the image on the graphical user interface (GUI).

#### 3.2. Vehicle detection

Vehicle detection is to isolate vehicles from other objects in the camera view. Several algorithms can be used, and one of the best is that based on utilizing a deep neural network because of its high recognition accuracy and fast time of detection. However, as it is not a main part of this work, we used project [21], which is based on faster R-CNN with VGG16 network to detect vehicle. The model is capturing

multiple moving objects from a video camera, and modified it to select only vehicle. We took several photos of several cars with different types of colors, different license plates, and different orientations. Figure 2 shows a sample of cars taken by the camera.



Figure 2. Sample of vehicle (car) detection with newer Iraqi license plate

### 3.3. License plate extraction

At this stage, the system extracts the vehicle license plate data. At this stage, an image is taken, the region with the plate number is found, its position is determined, and the cropped license plate is returned. The algorithm is described below. When applied this process on input image of left car in Figure 2, the results is shown in Figure 3

- Input**     **RGB Image**  
**Output**   **Binary image**  
**Step 1:**    Take the detected vehicle image.  
**Step 2:**    Convert to grayscale.  
**Step 3:**    Utilize both the edge and the Sobel operator [22], to compute the threshold value.

$$edge(a,b) = \begin{cases} \text{Max}(a,b) & \text{if } \text{grade}(a,b) > a * \text{Max}(a,b) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

- Step 4:**    The threshold value needs to be tuned. Then, use the edge again in order to get the binary mask that contains the segmented license plate (in this work we have used a 0.5 fudge factor to change the threshold level from 0.1-0.99) and generate the binary gradient mask.  
**Step 5:**    Dilate the binary gradient mask by creating two perpendicular linear structuring elements using the morphological structuring element in order to create a non-flat structuring element [23].

$$A \circ B = \cup \{ (B)_z \mid (B)_z \subseteq A \} \quad (2)$$

A vertical edge thickness ( $s_{90}$ ) and a horizontal edge thickness ( $s_0$ ). The binary gradient mask is used to find the lines that have high contrast within the image. These lines usually do not quite delineate the object's outline. In comparison to the original image, there are several gaps in the lines that surround the object inside the gradient mask. However, these gaps can be eliminated by dilating the Sobel image by using the linear structuring elements.

- Step 6:**    Fill the interior gaps by using the Flood-Fill operations [24] on binary images. This operation is used to remove an irrelevant artifact from images. (After dilating the Sobel image, the object outline will be quite nice; however, there are still many holes inside the object. Thus, we used the Flood-Fill operations to fill these holes.  
**Step 7:**    The edge is thickened using a diamond of size 10.

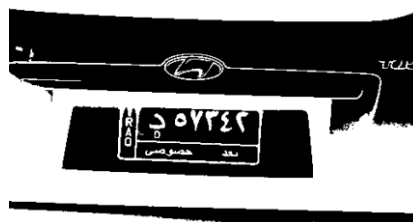


Figure 3. Isolat license plate region

### 3.4. Characters detection, segmentation and recognition

This stage is used to detect and recognized character through several of process that includes: number detection and recognition; one English character digit detection and recognition; governorate detection and recognition; and color recognition.

#### 3.4.1. Numbers detection and recognition

This part is used to detect and recognize Arabic numerals. The process takes the cropped image of the plate number and returns digits in the form of an array. The proposed algorithm is described below. The result is shown in Figure 4.

**Input** Binary Image

**Output** Text (Number + Character)

**Step 1:** Preprocess the license plate image.

1. Convert an RGB image to a grayscale image.
2. Use Otsu's method to apply multilevel image thresholds.
3. Use specified quantization levels to quantize the image and output values.
4. Convert the label matrix into an RGB image.
5. Convert an image to a binary image based on the threshold.
6. Resize the image to a (512×1024) image.

**Step 2:** Determine the flat morphological structuring element by using the "Morphological structuring element [15].

1. Determine an array size (m n).
2. Use conditional statements to verify the following:
  - o If the sum of the array elements (SAE) < (m\*n - SAE), then dilate the image.
  - o If the sum of the array elements (SAE) ≥ (m\*n - SAE), then erode the image.
3. Create the label matrix using the connected objects.

**Step 3:** Conduct non-number elimination to remove the objects that are either too high or too low. Take the labeled image as the input and remove the unwanted objects.

1. Compute the maximum possible range ( $max_r$ ,  $max_c$ ), the minimum possible range ( $min_r$ ,  $min_c$ ), and the Difference ( $D_r$ ,  $D_c$ )
2. Use conditional statements to verify the following:  
If  $D_c(i) < 30$  OR  $D_c(i) > 175$ , then remove the index segment from the labeled image.
3. Use the formed histogram bins to remove objects that are too high and too low.

**Step 4:** Take the labeled image (L) and return the (N) 2D images stored in t, where N is the number of labeled objects in L.

1. Compute the maximum possible range ( $max_r$ ,  $max_c$ ), the minimum possible range ( $min_r$ ,  $min_c$ ), and the Difference ( $D_r$ ,  $D_c$ ).
2. Crops the number based on the location and dimensions described in the range values.
3. Resize the cropped number to (128×70).
4. Use the loop function to determine how many max components to retain.

**Step 5:** Number Classification. Classify the 2D images based on the masks stored in the "Masks" table.

1. Load the "Masks" table.
2. Use the correlation coefficient (corr2)[25], which is a number that represents the similarity between two images associated with their specific pixel intensity.

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (3)$$

Where  $\bar{A} = \text{mean2}(A)$  and  $\bar{B} = \text{mean2}(B)$ .

3. If a hole is detected in the 2D image, take the labeled image (L) and the detected number array, and insert zeroes in the array based on the difference in the objects in the image L.

- Compute ( $max_c$ ) and ( $min_c$ ).
- Compute the maximum elements of an array ( $mx$ ).
- Compute the width:

$$W = max_c - min_c \quad (4)$$

- Compute the average width:

$$AW = \sum \frac{W}{mx} \quad (5)$$

- Insert zeroes if the difference is greater than 0.65 and the next number is

- not one.
- Insert zeroes if the difference is greater than 0.85 and the next number is one.
  - Output the recognized image.



Figure 4. License plate numbers and characters recognition

### 3.4.2. One digit english character detection and recognition

This stage uses the same steps as in section (4.1); however, there are some differences in this part. Step 3 eliminates the non-English characters. In step 5, the 2D images of the characters are compared with the masks stored in the "Masks" table, and point 3 is neglected.

### 3.4.3. Car type classification

Car type classification. To achieve this, in this part, the system recognizes the color of the plate. The algorithm is described as follows:

- Input** Binary Image  
**Output** Text
- Step 1:** Subtract the image from mask.  
**Step 2:** Use the median filter to filter out the noise,  
**Step 3:** Convert the image to a binary image.  
**Step 4:** Remove pixels that are less than 300px.  
**Step 5:** The connected components in the image are labeled.  
**Step 6:** Determine the image regions' properties. By this, we can perform image blob analysis and get a set of properties for each labeled region.  
**Step 7:** Load the "color" table.  
**Step 8:** Compare the RGB value and determine the car type.

## 4. RESULTS AND DISCUSSION

The AILPR software has been tested in order to investigate its operations during real-time video streaming, car detection, and extracting the license plate in order to recognize the license plate numbers. The system has been designed using MATLAB R2020b. Figure 5 shows the GUI of the proposed system software. The test images were taken for various Iraqi license plates (old style and new style) and under various illumination and positioning conditions. First, we tested the extraction of the vehicle license plate data.

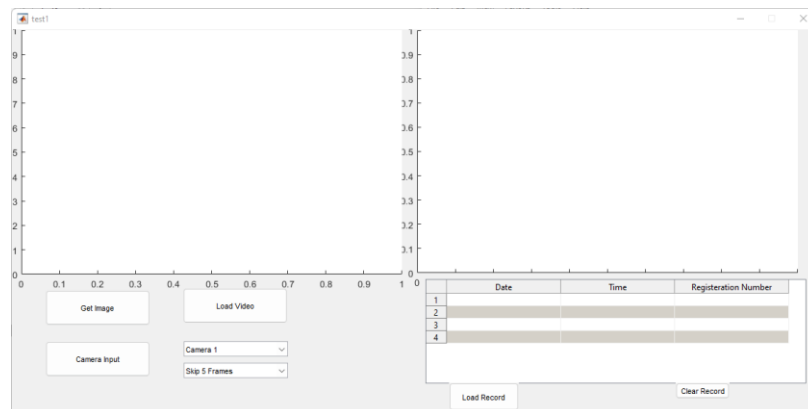


Figure 5. GUI of the proposed system software

In this test, we investigate the extraction accuracy for different types of vehicles, different colors, old and new plate styles, and different illumination conditions. The system is tested in street for real investigation where we put camera at distance near 10-meters capture. From Figure 6, two types of cars (public and private) with different angles have been captured, the first one has new style private license plate (Figure 6(a)) and the second one has old public style license plate (Figure 6(c)). The results in (Figure 6(b)) and (Figure 6(d)) show that the recognition date and recognized license plate number which lies in below of image and it show that the system successfully recognized both license plates correctly.

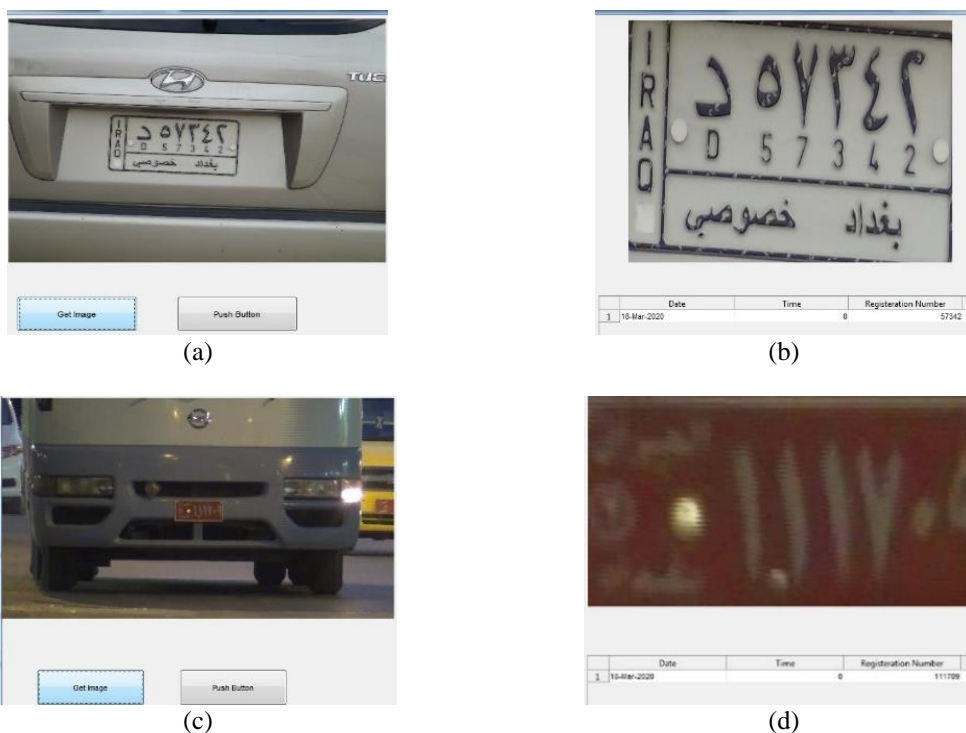


Figure 6. License plate extraction for different vehicles in different positions and illuminations. Streamed video (left), (a) private car have new style license plate, (b) recognition result for license plate number, (c) small bus having old style public license plate, and (d) recognition result of license plate number

As shown in Figure 6, the system accurately extracted the license plate with good results. The plate has good dimensions, and the cropping successfully isolates the plate close to its edges. The second test determines the accuracy of the system when recognizing license plate numbers. Like in the previous test, we test different types of vehicles, different colors, old and new plate styles, and different illuminations. Figure 4 shows some of the license plate extraction.

The system detects and recognizes the license plate numbers of all vehicles in different situations (different styles, colors, and angles) efficiently with no errors, as shown in Figure 7, which are includes: clear old private style license plate Figure 7(a), tilted captures of old public style license plate Figure 7(b), low illumination public style license plate Figure 7(c), and tilted captures of New Private style License plate Figure 7(d). However, the system has errors in some cases, which can happen when there are some aspects that influence the detection and recognition process, such as unclear numbers related to some mud or dirt that hides or distorts parts of the LP. In addition, the processing time is very high, which takes about 5–9 msec. To test the system's accuracy, we put it on branch street for one day and captured both the car image and the license plate, as shown in Table 2.

From Table 2, we investigate a real-time recognition system in real operation. From the result, the system detects different Iraqi license plates depending on its types (that is explained in Table 1), in which 326 license plate cars are detected, and 319 of them are detected correctly, while 7 of the LPs are detected wrongly. However, it can be noticed that most wrong recognition is in the new style Iraqi LP. This is due to a number of factors, including non-clear details related to being covered with mud and not having clear details that do you damage in LP. From these results with a variety of styles (in Iraq there are 11 types of LP with several styles and colors, some include just Arabic numbers and characters, some include Arabic and English



characters, some come with a total black or red or blue color background with white characters, some with a yellow color for background and black color for characters, and the new style has a part with color which is actually very hard for any license plate recognition system to detect all accurately, with high accuracy reaching up to 97.85%. Hence, the contribution of the proposed model that makes it overcome all the Iraqi license plate recognition challenges can be described as follows:

- Color and style are independent (all LPs are grayscale and detect desired number region based on Arabic number location).
- Ignore non-Arabic numbers. In this method, we proposed a region detection that starts based on the first Arabic number.
- It combined two techniques: fast object detection and model-based CNN to detect vehicle; and template matching to templet matching to recognize LP.

These which in combination give a benefit from high accuracy in detection and in very fast processing time. Table 3 show the compaction between our method and some recent related works. In summery the system detects all license plates and it recognize the LP in of car efficiently with high accuracy reach up to 97.85.



Figure 7. License plate recognition, (a) clear old private style license plate, (b) tilted captures of old public style license plate, (c) low illumination public style license plate, and (d) tilted captures of new private style license plate

Table 2. Proposed ALPR system recognition results

Vehicle type	Number of each license plates types detected		Correct recognition		Wrong recognition	
	Old style	New style	Old style	New style	Old style	New style
Private	7	284	7	279	0	5
Public	1	12	1	12	0	0
Trade	0	0		-----	0	0
Governmental	3	16	3	14	0	2
Temporary customs entry	0	0		-----		-----
Agriculture	0	0		-----		-----
Temporary permit	3	---	3		0	7
Total cars detected	326	Total number of LP recognized correctly	319	Total number of LP wrong recognized		7

Table 3. Proposed ALPR system recognition results

Recent works	Method	Results
Li <i>et al.</i> [12]	Multi-layer perceptrons for plate detection and bounding box regression, and RNNs with CTC for plate recognition	96.13% end-to-end performance and 98.15% in detection-only performance for carflag-large dataset processing time about 300ms. Achieved 95.29% in end-to-end performance and 99.56 % in detection-only performance for aolp dataset with processing time about 400ms
Xie <i>et al.</i> [13]	CNN-based MD-YOLO framework	Achieve 98.32% in accuracy when uses UCSD dataset and 97.38% when used PKU dataset and the processing time is (530+140) ms when using CPU and (5+2.7) when using GPU
Srinu <i>et al.</i> [14]	Model-based template matching	Achieve 96.6% accuracy with good prediction. Has shorter time required for recognition which is about 35ms
Min <i>et al.</i> [15]	Two models, a k-means++ clustering algorithm is used to determine the optimal range for the number of plate candidate boxes and their sizes. Second, the YOLOv2 model's structure and depth	Achieving a 98.86% in both precision and recall. The processing time is about 31.06 second
Alhamza and Alaythawy [16]	KNN	Achieving a 90% in accuracy
Zandi and Rajabi [17]	Two network, first CNN is the YOLOv3 network to finds the Iranian license plate in the input image. The second CNN is a faster R-CNN, to recognizes and sorts the characters in the found license plate	Got 98.97% recall, 99.9% precision, and 98.8% accuracy. The processing time 23ms
Proposed model	Two algorithm, first faster R-CNN with VGG16 network to detect vehicle. The second is template matching with modified algorithm to recognizes and sorts the characters in the found license plate	Achieve 100% in all uniform clear license plate (in experimental test), and achieve high accuracy reach up to 97.85% in real time in real operation in street

## 5. CONCLUSION

In this paper, the software of an ALPR system is designed. First, the real-time video is streamed to the system. Then, the system detects the license plate position and extracts the data via multipreprocessing and segmentation operations. Finally, a four-step process is used to recognize four aspects: Arabic numerals, one English character, Arabic characters, which represent the governorate, and color, which specifies the vehicle type. The plate data passes through multiple processes to obtain the license plate number, governorate, and type. This system was tested using different vehicles with different license plate types (326 images) in different positions. The test results prove that the proposed model and the overall process are successful. This system successfully detects and extracts license plates from the background with higher accuracy, reaching up to 97.85%.




## REFERENCES

- [1] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies," *Journal of Modern Transportation*, vol. 24, no. 4, pp. 284–303, Dec. 2016, doi: 10.1007/s40534-016-0117-3.
- [2] P. A. Hancock, I. Nourbakhsh, and J. Stewart, "On the future of transportation in an era of automated and autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7684–7691, Apr. 2019, doi: 10.1073/pnas.1805770115.
- [3] C. Patel, D. Shah, and A. Patel, "Automatic number plate recognition system (ANPR): A survey," *International Journal of Computer Applications*, vol. 69, no. 9, pp. 21–33, May 2013, doi: 10.5120/11871-7665.
- [4] S. Mahalakshmi and R. S. Kumar, "Smart toll collection using automatic license plate recognition techniques," in *Communications in Computer and Information Science*, 2018, pp. 34–41, doi: 10.1007/978-981-13-0755-3\_3.
- [5] K. Mohammad, S. Agaian, and H. Saleh, "Arabic license plate recognition system," *Journal of Signal Processing Theory and Applications*, vol. 1, pp. 20–42, 2013, doi: 10.7726/jspta.2013.1002.
- [6] M. J. Rahman, S. S. Beauchemin, and M. A. Bauer, "License plate detection and recognition: An empirical study," in *Science and Information Conference*, 2020, pp. 339–349, doi: 10.1007/978-3-030-17795-9\_24.
- [7] S. S. Omran and J. A. Jarallah, "Iraqi car license plate recognition using OCR," in *2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT)*, Mar. 2017, pp. 298–303, doi: 10.1109/NTICT.2017.7976127.
- [8] A. H. M. Alaidi, S. A. A. A. Alsaïdi, and O. H. Yahya, "Plate detection and recognition of iraqi license plate using KNN algorithm," *Journal of Education College Wasit University*, vol. 1, no. 26, pp. 449–460, Jan. 2017, doi: 10.31185/eduJ.Vol1.Iss26.102.
- [9] Wikiwand, "Vehicle registration plates of Iraq," *Wikiwand*, 2022. [https://www.wikiwand.com/en/Vehicle\\_registration\\_plates\\_of\\_Iraq](https://www.wikiwand.com/en/Vehicle_registration_plates_of_Iraq) (accessed Sep. 18, 2022).
- [10] G. K. Ali, "Developing recognition system for new iraqi license plate," *Tikrit Journal of Engineering Sciences*, vol. 25, no. 1, pp. 7–10, Mar. 2018, doi: 10.25130/tjes.25.02.
- [11] A. Atanassov, "Advanced software architecture of an automatic vehicle number plate recognition system," *Journal of the University of Chemical Technology and Metallurgy*, vol. 47, no. 1, pp. 77–82, 2012.




- [12] H. Li, P. Wang, and C. Shen, "Towards end-to-end car license plates detection and recognition with deep neural networks," Sep. 2017.
- [13] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, Feb. 2018, doi: 10.1109/TITS.2017.2784093.
- [14] M. V. Srinu, V. R. Morla, K. V. P. Baditha, V. Kumari. S, and S. Maddimsetti, "License plate recognition system using matlab GUI for surveillance," *International Journal of Engineering & Technology*, vol. 7, no. 2.7, pp. 1008–1016, Mar. 2018, doi: 10.14419/ijet.v7i2.7.11676.
- [15] W. Min, X. Li, Q. Wang, Q. Zeng, and Y. Liao, "New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification," *IET Image Processing*, vol. 13, no. 7, pp. 1041–1049, May 2019, doi: 10.1049/iet-ipr.2018.6449.
- [16] D. A. Abd Alhamza and A. D. Alaythawy, "Iraqi license plate recognition based on machine learning," *Iraqi Journal of Information & Communications Technology*, vol. 3, no. 4, pp. 1–10, Dec. 2020, doi: 10.31987/ijict.3.4.94.
- [17] M. S. Zandi and R. Rajabi, "Deep learning-based framework for Iranian license plate detection and recognition," *Multimedia Tools and Applications*, vol. 81, no. 11, pp. 15841–15858, May 2022, doi: 10.1007/s11042-022-12023-x.
- [18] B. Li, Z. Y. Zeng, H. L. Dong, and X. M. Zeng, "Automatic license plate recognition system," *Applied Mechanics and Materials*, vol. 20–23, pp. 438–444, Jan. 2010, doi: 10.4028/www.scientific.net/AMM.20-23.438.
- [19] H. Padmasiri, J. Shashirangana, D. Meedeniya, O. Rana, and C. Perera, "Automated license plate recognition for resource-constrained environments," *Sensors*, vol. 22, no. 4, pp. 1–29, Feb. 2022, doi: 10.3390/s22041434.
- [20] MathWorks, "Motion-based multiple object tracking," 2022. <https://www.mathworks.com/help/vision/ug/motion-based-multiple-object-tracking.html> (accessed Aug. 21, 2022).
- [21] H. K. Leung, X.-Z. Chen, C.-W. Yu, H.-Y. Liang, J.-Y. Wu, and Y.-L. Chen, "A deep-learning-based vehicle detection approach for insufficient and nighttime illumination conditions," *Applied Sciences*, vol. 9, no. 22, pp. 1–27, Nov. 2019, doi: 10.3390/app9224769.
- [22] C.-C. Zhang and J.-D. Fang, "Edge detection based on improved sobel operator," in *Proceedings of the 2016 International Conference on Computer Engineering and Information Systems*, 2016, pp. 129–132, doi: 10.2991/ceis-16.2016.25.
- [23] K. Miura, "Bioimage data analysis".
- [24] P. M. Chu, S. Cho, K. Huang, and K. Cho, "Flood-fill-based object segmentation and tracking for intelligent vehicles," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, pp. 1–11, Nov. 2019, doi: 10.1177/1729881419885206.
- [25] C. Minwalla, E. Shen, P. Thomas, and R. Hornsey, "Correlation-based measurements of camera magnification and scale factor," *IEEE Sensors Journal*, vol. 9, no. 6, pp. 699–706, Jun. 2009, doi: 10.1109/JSEN.2009.2021197.

## BIOGRAPHIES OF AUTHORS






**Sukaina Sh Altyar**    has received a Bachelor of Computer Science degree from the University of Baghdad, Iraq in 2008, and a Master of Computing science degree from Baghdad science University in 2015. She is a lecturer at the College of Education for Pure Science Ibn-Al-Haitham, Department of Computer, University of Baghdad since 2015. Her current research interests include the areas of image processing and artificial intelligent. She can be contacted at email: sukaina.s.m@ihcoedu.uobaghdad.edu.iq.



**Samera Shams Hussein**    received a Bachelor of Computer Science degree from the University of Baghdad, Iraq in 2008, and a Master of Computing degree from Information Institute for postgraduates studies, Iraq in 2013, She is a lecturer at the College of Education for Pure Science Ibn-Al-Haitham, University of Baghdad since 2009. Her current research interests include the areas of image processing and artificial intelligent. She can be contacted at email: Samera.s.h@ihcoedu.uobaghdad.edu.iq.



**Lubab Ahmed Tawfeeq**    received a Bachelor of Computer Science degree from the University of Baghdad, Iraq in 2003, and a Master of Computing degree from Arab Academy for Science Technology and Maritime Transport, Egypt in 2013, She is a lecturer at the College of Education for Pure Science Ibn-Al-Haitham, University of Baghdad since 2005. Her current research interests include the areas of image processing and machine learning. She can be contacted at email: lubab.a.t@ihcoedu.uobaghdad.edu.iq.