P-ISSN 1991-8941
E-ISSN 2706-6703
Vol. 20, No. 1 (2026)

Journal of University of Anbar for Pure Science
(JUAPS)

journal homepage: juaps.uoanbar.edu.iq

Open Access

Original Paper

A new procedure for automatic vehicle license plate recognition monitoring system via CNN and RCNN and Arduino

Anwar H. Al-Saleh[1]**, Mohanad Ali Meteab Al-Obaidi**[1*]**, Shajan Mohammed Mahdi**[2]**, Ali Abid D Al-Zuky**[3]

[1]Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq
[2]Department of Computer, College of Education, Mustansiriyah University, Baghdad, Iraq
[3]Department of Physics, College of Science, Mustansiriyah University, Baghdad, Iraq

ARTICLE INFO

Received: 09/06/2025
Accepted: 11/08/2025
Available online: 23/03/2026
April Issue
10.37652/juaps.2025.161328.1428

CITE @ JUAPS

Corresponding author
Mohanad Ali Meteab Al-Obaidi
neros2210@uomustansiriyah.edu.iq

ABSTRACT

Ensuring security for vehicle access control in vital institutions, factories, and residential complexes has become a pressing concern. To address this, leading global logistics companies have focused on enhancing security infrastructure through vehicle license plate recognition systems. This research employs two algorithms: Region-based Convolutional Neural Network (RCNN) and Convolutional Neural Network (CNN). Both deep learning models are essential and are widely applied for vehicle detection and license plate recognition in image processing tasks. The system integrates an Arduino microcontroller, an ultrasonic sensor, and a computer-connected camera to capture images of vehicles at checkpoints. These images are then compared with a database of authorized vehicles. Arduino provides an efficient and adaptable platform for object detection and classification across different domains, including robotics, surveillance, and automation, enabling the development of more advanced systems in the future. The system achieved an accuracy rate of over 98%, demonstrating its reliability and effectiveness. This study makes a significant contribution by providing a robust security strategy.

Keywords: *Arduino UNO, CNN, license plate, OCR, RCNN, Ultrasonic sensor, Vehicle, VLPRS*

1 INTRODUCTION

Today's world has seen rapid developments in economic, industrial, social, and urban aspects. Therefore, security infrastructure has emerged as a crucial tool for civil development. Every day, we look for innovative ways to find solutions that ensure the safety and well-being of individuals and communities. One of these pioneering security developments is the construction of a vehicle license plate recognition system (VLPRS), which relies on region-based convolutional neural network (RCNN) and convolutional neural network (CNN) algorithms [1–3], and utilizes a database to store images of vehicle registration plates. Controlling the movement and entry of vehicles and managing them electronically in security, industrial, and military complexes presents a challenge. Therefore, developers and engineers from all over the world have built and developed numerous algorithms and technologies that are related to each other. These algorithms rely on multiple variables, such as detecting image edges, matching templates, and colors, to detect and recognize car plates more easily and efficiently than other methods. Different techniques, such as Girshick's RCNN algorithm, can solve the common problem of recognizing license plate numbers from car images in computer vision [4–6]. Characters from the detected plate area are recognized by the RCNN algorithm [7], designed for deep learning-based object detection [8]. In the second stage, CNN is the most common algorithm for image classification and object detection methods, and is considered very suitable due to its unique structure [9–11]. The term "convolutional" describes it, as convolutional layers process the input data and capture the spatial relationships between pixels in the image [12]. CNN uses convolutional layers to process the input image and detect various features. Each convolutional layer applies filters to the inputs [13]. Pooling layers are used to sample the outputs of these filters, reducing data dimensionality and computational cost [14]. Next, they produce the final classification output by passing the pooling layer's output through one or more fully connected layers. In this paper, the primary objective of the proposed RCNN is to discern and extract license plates from the frontal view of vehicles [15], distinguishing them from the surrounding vehicle front shape. Similarly, the main goal of the proposed CNN is to classify and identify vehicle license plates. Notably, this model greatly improves the efficiency of accurately detecting and classifying these license plates. Given that Iraqi vehicles carry license plates with a specificity similar to those used in other countries around the world, this study proposes a structure to identify the license plate (LP). The panels feature English letters (A, B, C,...,Z) and Arabic numbers (0–9), which serve as abbreviations for the names of the Iraqi governorates. We propose the RCNN algorithm for LP recognition and the CNN algorithm for vehicle recognition. The process involves detecting the LP (using RCNN), extracting it from the vehicle image, and then comparing the extracted image of the vehicle plate with the database of registration plates of vehicles permitted to enter the site (using CNN). This process takes into account sufficient factors affecting the image, such as the car's deflection, the presence of dust on the plate, and changes in lighting intensity, day or night. The investigation has a significant impact on military facilities and critical factories because it provides an effective safety mechanism. Additionally, it provides extensive vehicle tracking and access management features by utilizing license plate identification technology and machine learning algorithms such as RCNN and CNN. Arduino, coupled with cameras and ultrasonic sensors, allows for effective object identification and categorization, opening avenues for the design of current systems that include robots, monitoring, and automation.

2 RELATED WORK

Numerous studies have examined the recognition of characters [16–18], numbers [19–21], and license plates [22–25] using conventional methods, machine learning, and deep learning techniques. A new technique for character recognition and license plate detection was provided in [26]. Their approach blends a backpropagation neural network (BPNN) with feature extraction methods. Three sets of algorithms are used in the feature extraction process: extracting features using the horizontal traverse density (HTD) and vertical traverse density (VTD) methods, edge distance features, and training feature vectors using BPNN, which exhibited 97.7% recognition accuracy.

Zou Y. et al. [27] presented a sophisticated, three-component model for recognizing license plates. First, Xception, MobileNetV3, and spatial attention techniques were used to extract license plate features. Bi-LSTM was used to identify the character locations on license plates. A 1D-attention system was used to extract character features, enhancing the beneficial ones and suppressing the irrelevant ones.

Ghida Y. et al. [28] created two databases, one with 2000 images of Iraqi city names and the other with 1200 images, to identify Iraqi vehicle license plates. They applied an algorithm to recognize Iraqi license plates based on deep learning techniques, namely the Convolutional Neural Networks (CNNs) to classify images, numbers, and Arabic letters. They used methods to segment numbers, letters, and words from pictures of a car license plate to train the CNN algorithm to identify their features. They achieved an overall recognition accuracy of 98%.

H. Nguyen [29] presented a novel approach for detecting license plates that relies on lightweight deep convolutional neural networks. Through the development of a hierarchical generation module that extracts high-resolution features from input images and a feature enhancement module that fuses these features with basic region features, the method demonstrated a 72 ms inference time and an accuracy rate of 98.68%.

Tharaa and Faisal [30] presented a developed system in which they used deep learning techniques to identify the license plate number and logo of vehicles in Jordan. They used the YOLOv3 model for license plate detection, character recognition, and vehicle logo detection. They trained the visual engineering group on how to recognize car logos. They used the average accuracy measure to evaluate the YOLOv3 model and obtained results for precision, recall, and F1 measurements of 95.3%, 99.5%, and 97.4%, respectively.

RCNN and CNN algorithms with common hardware (Arduino, camera, and Ultrasonic sensor) are powerful building blocks for deep learning in image recognition systems. This system can efficiently detect and recognize LPs, even in difficult lighting and weather conditions. Algorithms have enabled robust, adaptable, integrated, low-cost, and energy-efficient solutions. Training deep learning models like RCNN and CNN on large datasets allows them to adapt to changes in LP design, fonts, and backgrounds.

3 HARDWARE SYSTEM IMPLEMENTATION REQUIREMENTS

This section elaborates on the tools and devices designed to create a sensor system and capture vehicle data, all of which are detailed below:

3.1 Arduino uno 3

The Arduino UNO 3 is a control board based on the ATmega328P. It has 14 digital input/output ports; 6 ports can be used as outputs and 6 ports as analog inputs. Additionally, it features a USB connection, a power socket, an ICSP header, and a reset button. It is widely used in most IoT applications, as well as in video capture applications by operating cameras via a computer, which exists in several versions like the Arduino Leonardo Board, Lilypad Arduino Board, Arduino UNO (R3), and Arduino Mega 2560 (R3) [31, 32].

3.2 Ultrasonic sensor

An ultrasonic sensor is a low-cost sensor for measuring the distance of moving objects, as it sends high-frequency ultrasonic waves to measure the distance relative to the object's location [33]. This sensor can detect objects up to 4 meters. It can be connected to the Arduino. It usually operates at a rated DC voltage of 5 volts and a rated current of 3 mA.

3.3 Logitech quickcam

A Logitech QuickCam camera with a resolution of 1280 x 1024 pixels was employed and connected directly to a computer via USB to provide high-resolution images of vehicles. This type of camera has an automatic light correction feature in natural colors.

3.4 Sensing process

The apparatus uses an ultrasonic sensor that is connected to the Arduino for detecting vehicles within a three-meters range, enabling the Arduino to send a signal to the computer. Figure 1 below illustrates this process.

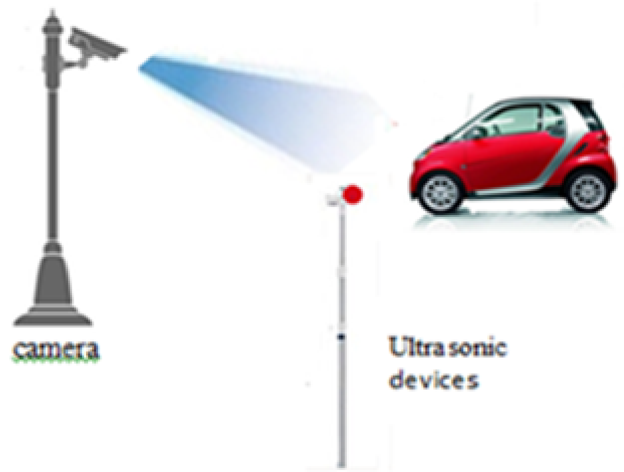

Fig. 1 Ultrasonic system with camera

A signal prompts the camera to capture a picture of the vehicle, the RCNN detects the vehicle's LP, then the CNN recognizes the LP characters and numbers to identify the vehicle. Figure 2 shows the steps of the proposed system.

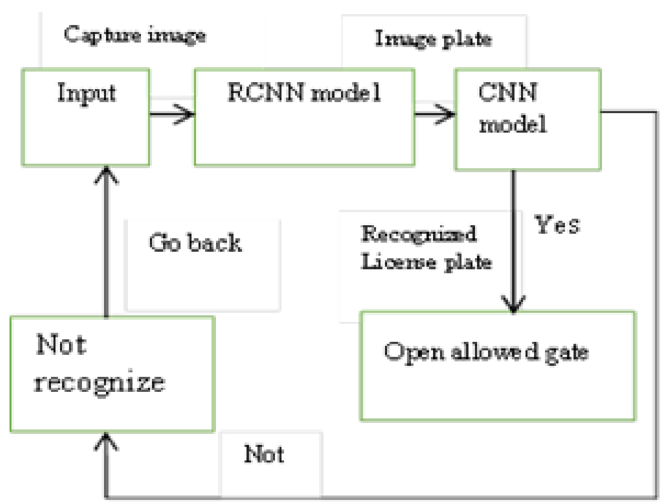

Fig. 2 The flow diagram of the proposed system

3.5 Proposed system architecture

This paper proposes a vehicle entry monitoring system (VLPRS) at locations that require continuous monitoring.

The VLPRS is designed and implemented in the following stages:

1. Creation (DB1): Create and collect the initial database (DB1) of all vehicles that have been scanned while passing through the checkpoint. This can be achieved by taking pictures of the vehicle and recording it while it is moving from different angles, distances, and lighting conditions. Figure 3 illustrates the process of creating this database.

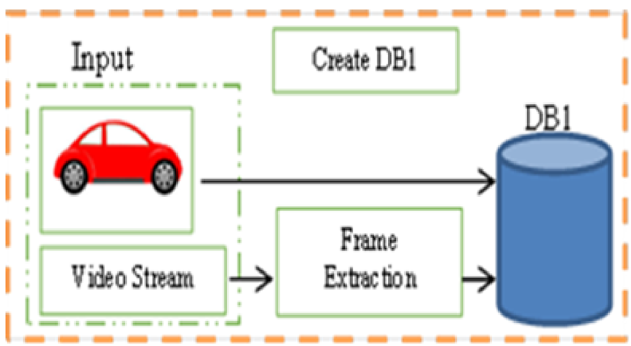

Fig. 3 The process of creating DB1

2. RCNN Training: Perform RCNN training to detect car LP in car images using the following steps:

   i Manually label car LPs in images in DB1 and save the labeling file (Lb-file).

   ii Use (Lb-file) and DB1 to perform the training process and obtain RCNN detector model (RCNN-det), then save this model, as shown in Figure (4).

3. Creation of (DB2): The following steps are used to create a second database (DB2) to detect license plates in images in (DB1):

   i Use (RCNN-det) to localize and extract the license plate images (IE) from (DB1) images.

   ii The IE is manually divided into four classes, one for each identified vehicle, and the results are then saved to DB2, as shown in Figure 5.

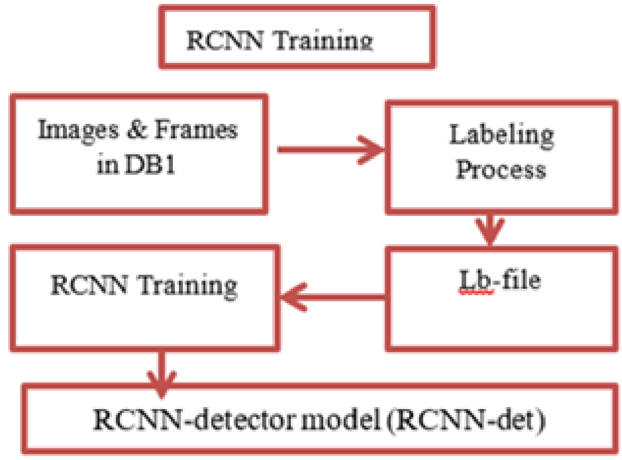

Fig. 4 The creating RCNN-det Model

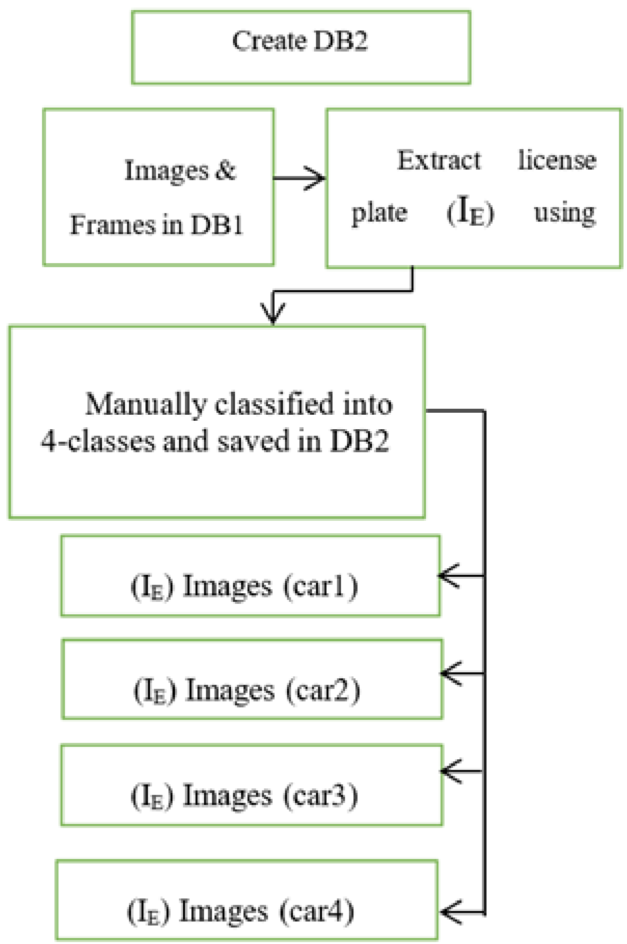

Fig. 5 The process of creating DB

4. VLPRS: At this stage, the system works as follows:
   - i Capture an image of the vehicle (I).
   - (a) The RCNN-det model is used to detect the license plates in (I), and extract the license plate image (IE).
   - (b) CNN-rec model classifies the license plate image (IE) into one of the four known classes or as unknown to the system.
   - iv When the device recognizes the vehicle license plate (IE), it sends an approval signal to open the gate for the vehicle, as shown in Figure 6.

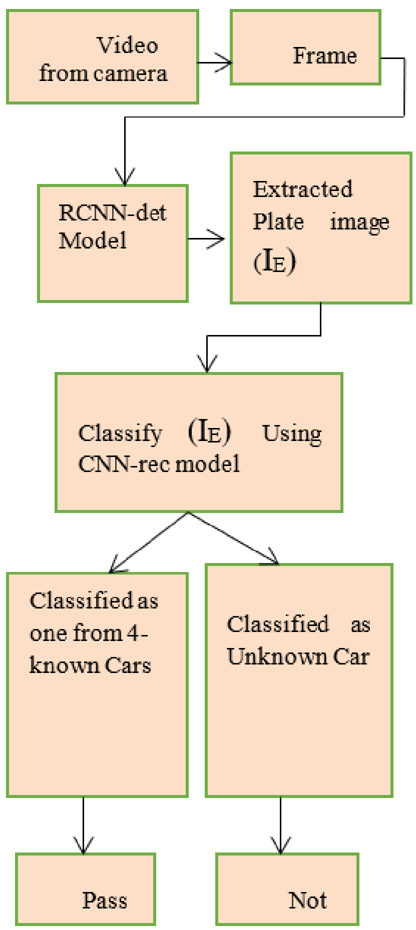

Fig. 6 The steps of VLPRS system

4 RCNN

This stage is the training stage where nine layers are used as shown below:

- i The image input layer refers to an input image size of 64x32 pixels with 3 color channels (RGB).
- ii The Convolution2 layer performs convolutional filtering on the input using 64 filters of size 5x5. The "Padding" parameter is set to "same" to pad the input with zeros to preserve the spatial dimensions
- iii The Batch Normalization layer: This layer normalizes the activations of the previous convolutional layer by subtracting the mean and dividing by the standard deviation.
- iv MaxPooling2D layer: Performs max pooling with a pool size of 3x3 and a stride of 3, which reduces the spatial dimensions while retaining the most important features.
- v ReLU layer: Applies the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity and enhances the network's ability to learn complex patterns.
- vi Repeat steps ii–v: To extract deeper features, add two more sets of convolutions, batch normalization, max pooling, and ReLU layers with different filter sizes (32 and 16).
- vii Fully connected layer: Connects all neurons from the previous layer to the output layer. The variable m, which typically represents the number of classes in the classification task, specifies the number of neurons in this layer.
- viii Softmax layer: This layer applies the Softmax activation function, which converts the output values to probabilities, representing the class probabilities for multi-class classification.
- ix Classification layer: Specifies the final classification layer that computes the cross-entropy loss and performs classification based on the predicted probabilities.

4.1 Options

The training options for the object detection model are set, including the optimizer, execution environment, and parameters, which control various aspects of the training process, such as the optimization algorithm, learning rate, batch size, and verbosity, allowing for customization based on the requirements of the model and the specific dataset:

i 'sgdm': Stochastic gradient descent with momentum optimizer. Uses mini-batches to update the network weights.

ii 'Momentum', 0.9000: Sets the momentum value to 0.9. Momentum helps accelerate gradient descent in the relevant direction and dampens oscillations during training.

iii 'Execution Environment', 'CPU': Specifies that the training should be performed on the CPU. Alternatively, you can use 'auto' or 'gpu' to leverage GPU acceleration, if available.

iv 'Mini Batch Size', 16: Sets the mini-batch size to 16. The network updates its weights during training by computing the average gradient from this number of samples.

v 'Initial Learn Rate', 1e-3: Sets the initial learning rate to 0.001. It determines the step size for weight updates during the initial training iterations.

vi 'Max Epochs', 20: Defines the maximum number of training epochs to perform. An epoch represents a complete pass through the entire training dataset.

vii 'Verbose Frequency', 100: Specifies that the display of the training progress should occur every 100 iterations. This provides periodic updates on the training loss and other metrics.

viii 'Verbose', 1: Enables verbose mode, which displays detailed information during the training process, including progress, mini-batch information, and validation metrics.

5 CNN TRAINING

This phase focuses on developing a CNN-Rec model to detect license plate classification (LP). The main goal is to identify and classify vehicle plates stored in the database (DB2). This model facilitates the final decision-making process on whether a vehicle is allowed or not to enter the facility. The following are the implementation steps:

i Image Input Layer: Indicates an input image size of 30x30 pixels with 3 color channels (RGB).

ii Convolution2 Layer: Performs convolutional filtering on the input with 8 filters of size 3x3. The 'Padding' parameter is set to 'same'.

iii Batch Normalization Layer: Normalizes the activations of the previous convolutional layer by subtracting the mean and dividing by the standard deviation.

iv Relu Layer: Applies the rectified linear unit (ReLU) activation function, which introduces non-linearity and enhances the network's ability to learn complex patterns.

v MaxPooling2D Layer: Performs max pooling with a pool size of 2x2 and a stride of 2, reducing the spatial dimensions while retaining the most significant features.

vi Repeat steps ii–v: To extract deeper features, two more sets of convolutional, batch normalization, ReLU, and max pooling layers are added with different filter numbers (16 and 32).

vii Fully Connected Layer: It connects all the neurons from the previous layer to the output layer. The number of classes in the classification task determines the number of neurons in this layer (4 classes).

viii Softmax Layer: It applies the softmax activation function. It converts the output values into probabilities, representing the class probabilities for multi-class classification.

ix Classification Layer: This specifies the final classification layer that computes the cross-entropy loss and performs the classification based on the predicted probabilities.

5.1 Options

'adam': It is an adaptive stochastic gradient descent with a momentum-based optimizer.

6 EXPERIMENTAL WORKS AND RESULTS

In this paper, vehicle images were collected using hardware equipment, namely an Arduino, a camera, and an ultrasonic sensor, as well as MATLAB-based algorithms. In addition, different environmental conditions, such as cloudy skies, lighting variation, rain, night, and different types of vehicles, were included. Two datasets (DB1 and DB2) were generated as shown in Figure 7 to sample the effects of these conditions. Two separate networks were trained for distinct tasks. The first network (RCNN), using (DB1), was trained to recognize and extract the LP of a vehicle within a vehicle image. The dataset (DB1) was randomly divided into a training set containing 75% of the data and a test set containing the remaining 25%. Then, the second network (CNN) was trained using the DB2 extracted from the first stage to recognize the vehicle based on the extracted LP information.

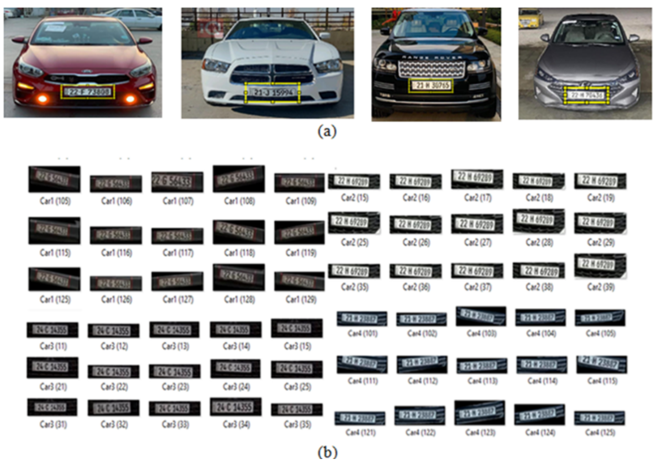

Fig. 7 Samples of creating databases of images from (DB1) for RCNN training.

The network was trained to extract the plate area from the input vehicle image, as shown in Figure 8. The results of the implementation of the presented system are as shown in Figures 8 and 9. The RCNN-det model was used to detect and extract the LP (IE), then the CNN-reg model was used to classify the (IE) into one of two classes: the first class is a known vehicle (allows passage), while the second class is an unknown vehicle (does not allow passage). Figure 9 shows the class classification section (allow), while Figure 10 shows the class classification section (do not allow).

Fig. 8 Extract the plate region from the image using RCNN-det model to create DB2.

Fig. 9 Samples of (DB2) for CNN training.

Results of performing the proposed system are shown in Figures 10, 11, and 12, where the RCNN-det model was used to detect and extract LP (IE), and then CNN-reg model was used to classify (IE) into one of two classes: the first class is a known vehicle (allow to pass), and the second-class is an unknown vehicle (not allowed to pass). Figure 11 illustrates the classification of the category (Allowing) passage, while Figure 12 illustrates the ratio and accuracy of the results obtained, where 50 different license plate images for each vehicle were covered. Finally, Figure 11 illustrates the classification of the category (Not Allowing) passage.

Fig. 10 Classify the category allowing passage.

Below is Figure 12 showing the accuracy of the results obtained in percentages.

Fig. 11 Classify the category not allowing

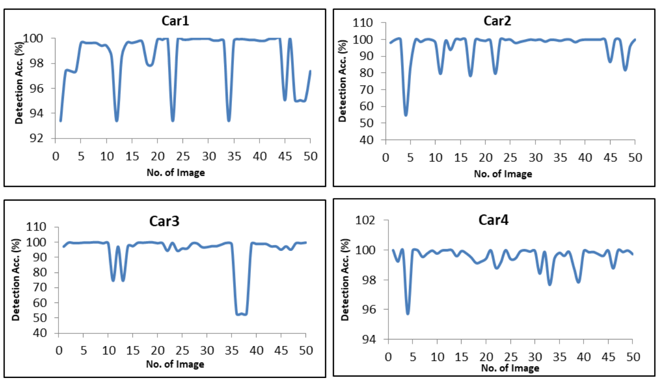

Fig. 12 the percentage and accuracy of the results obtained.

7 CONCLUSION

In this research paper, a region-based convolutional neural network (RCNN), a convolutional neural network (CNN), a camera, and an Arduino microcontroller were used to develop an efficient and reliable license plate recognition system. This solution combines the advanced image recognition capabilities of deep learning with the flexibility and control provided by Arduino, making it ideal for various privacy-focused enterprise security applications. Table 1 highlights the accuracy of the system implementation. The system accurately classified all license plate images stored in the test folder as vehicle class 1, where 1 indicates authorized vehicles entering the security location and any number below 1 indicates unauthorized vehicles. This license plate recognition system has proven to be a promising approach. With an impressive accuracy of over 98% and the remaining 2% representing blurred images, which render the vehicle subject to further security scrutiny, the study confirms the effectiveness and reliability of the proposed method in improving safety protocols for critical infrastructure and equipment.

Table 1 Comparison of Classification Results.

	Car1	Car2	Car3	Car4	NOT
Car1	1	0	0	0	0
Car2	0	0.88	0	0	0.12
Car3	0	0	1	0	0
Car4	0	0	0	1	0
NOT	0	0	0	0	1

Acknowledgement

The first author would like to thank the reviewers for providing useful suggestions, allowing for the improved presentation of this paper.

Funding source

No funds received.

Data availability

N/A

DECLARATIONS

Conflict of interest

The authors declare that no conflict of interest exists.

Consent to publish

N/A

Ethical approval

N/A

REFERENCES

[1] A J, Vikas, Samuel J, Bharanishwar, Srinivasan, Balan G, et al. Enhancing Public Safety through License Plate Recognition for Counter terrorism through Deep Learning Technique. In: 2023 4th International Conference on Signal Processing and Communication (ICSPC). IEEE; 2023. p. 96–100. 10.1109/icspc57692.2023.10125687

[2] Liu C. Methodology, application and platform for urban security planning and management. Journal of Infrastructure Policy and Development. 2024;8(4):1964. 10.24294/jipd.v8i4.1964

[3] Farooq MU, Ahmed S, Latif M, Jawaid D, Khan MZ, Khan Y. Detection and recognition of vehicle license plates using deep learning in video surveillance. International Journal of Computer Science & Network Security. 2022;22(11):121-6

[4] Ren J, Wang Y. Overview of object detection algorithms using convolutional neural networks. Journal of Computer and Communications. 2022;10(1):115-32

[5] Setiawan S, Sulistyowati DN, Machmud N. IMPLEMENTATION OF IMAGE PROCESSING IN THE RECOGNITION OF OFFICIAL VEHICLE LICENSE PLATES. JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer). 2023;9(1):23–29. 10.33480/jitk.v9i1.4181

[6] Yamini M. Number Plate Detection in an Image. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 2023;07(09):1–11. 10.55041/ijsrem25883

[7] Duan N, Cui J, Liu L, Zheng L. An End to End Recognition for License Plates Using Convolutional Neural Networks. IEEE Intelligent Transportation Systems Magazine. 2021;13(2):177–188. 10.1109/mits.2019.2898967

[8] D TD, V K. Deep Learning based Object Detection using Mask RCNN. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES). IEEE; 2021. p. 1684–1690. 10.1109/icces51350.2021.9489152

[9] Wu Z. Application of CNN Classic Model in Modern Image Processing. Journal of Advances in Engineering and Technology. 2024;1(3):1–6. 10.62177/jaet.v1i3.25

[10] Deng Z, Li A. Object Detection Algorithms Based on Convolutional Neural Networks. Highlights in Science, Engineering and Technology. 2024;81:243–251. 10.54097/vyfg4e34

[11] Yadav DK, Kumari N, Harron S. Retracted: Advances in Convolutional Neural Networks for Object Detection and Recognition. In: 2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC). IEEE; 2024. p. 1–6. 10.1109/icocwc60930.2024.10470695

[12] Pajankar A, Joshi A. In: Convolutional Neural Networks. Apress; 2022. p. 261–284. 10.1007/978-1-4842-7921-2_14

[13] Hakim H, Fadhil A. Survey: Convolution Neural networks in Object Detection. Journal of Physics: Conference Series. 2021;1804(1):012095. 10.1088/1742-6596/1804/1/012095

[14] Andrearczyk V, Whelan PF. Using filter banks in Convolutional Neural Networks for texture classification. Pattern Recognition Letters. 2016;84:63–69. 10.1016/j.patrec.2016.08.016

[15] Chen L, Li S, Bai Q, Yang J, Jiang S, Miao Y. Review of Image Classification Algorithms Based on Convolutional Neural Networks. Remote Sensing. 2021;13(22):4712. 10.3390/rs13224712

[16] Mohamad H, Hashim S, Al-Saleh A. Recognize printed Arabic letter using new geometrical features. Indonesian Journal of Electrical Engineering and Computer Science. 2019;14(3):1518. 10.11591/ijeecs.v14.i3.pp1518-1524

[17] Mirza NM. Printed Arabic Characters Recognition Based on Minimum Distance Classifier Technique. Iraqi Journal of Science. 2018;59(2A):762-70. 10.24996/ijs.2018.59.2A.14

[18] Amara M, Smairi N, Mnasri S, Zidouri A. Revitalizing Arabic Character Classification: Unleashing the Power of Deep Learning with Transfer Learning and Data Augmentation Techniques. Arabian Journal for Science and Engineering. 2024;49(9):12791–12815. 10.1007/s13369-024-08818-9

[19] Jabde M, Patil C, Mali S, Vibhute A. In: Comparative Study of Machine Learning and Deep Learning Classifiers on Handwritten Numeral Recognition. Springer Nature Singapore; 2023. p. 123–137. 10.1007/978-981-19-8094-7_10

[20] Jabde M, Patil C, Mali S, Vibhute A. Comparative Study of Machine Learning and Deep Learning Classifiers on Handwritten Numeral Recognition. In: International Symposium on Intelligent Informatics. Singapore: Springer; 2023. p. 123-37. 10.1007/978-981-19-8094-7$_1$0

[21] Ahmad I. A Hybrid Rule-Based and Machine Learning System for Arabic Check Courtesy Amount Recognition. Sensors. 2023;23(9):4260. 10.3390/s23094260

[22] Islam D, Mahmud T, Chowdhury T. An efficient automated vehicle license plate recognition system under image processing. Indonesian Journal of Electrical Engineering and Computer Science. 2023;29(2):1055. 10.11591/ijeecs.v29.i2.pp1055-1062

[23] Shi H, Zhao D. License Plate Recognition System Based on Improved YOLOv5 and GRU. IEEE Access. 2023;11:10429–10439. 10.1109/access.2023.3240439

[24] Pham TA. Effective deep neural networks for license plate detection and recognition. The Visual Computer. 2022;39(3):927–941. 10.1007/s00371-021-02375-0

[25] Pattanaik A, Balabantaray RC. Enhancement of license plate recognition performance using Xception with Mish activation function. Multimedia Tools and Applications. 2022;82(11):16793–16815. 10.1007/s11042-022-13922-9

[26] Xie F, Zhang M, Zhao J, Yang J, Liu Y, Yuan X. A Robust License Plate Detection and Character Recognition Algorithm Based on a Combined Feature Extraction Model and BPNN. Journal of Advanced Transportation. 2018;2018:1–14. 10.1155/2018/6737314

[27] Zou Y, Zhang Y, Yan J, Jiang X, Huang T, Fan H, et al. A Robust License Plate Recognition Model Based on Bi-LSTM. IEEE Access. 2020;8:211630–211641. 10.1109/access.2020.3040238

[28] Abbass GY, Marhoon AF. Car license plate segmentation and recognition system based on deep learning. Bulletin of Electrical Engineering and Informatics. 2022;11(4):1983–1989. 10.11591/eei.v11i4.3434

[29] Nguyen H. An Efficient License Plate Detection Approach Using Lightweight Deep Convolutional Neural Networks. Advances in Multimedia. 2022;2022:1–10. 10.1155/2022/8852142

[30] Aqaileh T, Alkhateeb F. Automatic Jordanian License Plate Detection and Recognition System Using Deep Learning Techniques. Journal of Imaging. 2023;9(10):201. 10.3390/jimaging9100201

[31] Zou L, Zhao M, Gao Z, Cao M, Jia H, Pei M. License Plate Detection with Shallow and Deep CNNs in Complex Environments. Complexity. 2018;2018(1). 10.1155/2018/7984653

[32] Mahdi SM, Al-Obaidi MAM, Mahdi RH. Comparative study of Arduino Types and Raspberry Pi with Programming Languages. Journal La Multiapp. 2023;3(5):250–255. 10.37899/journallamultiapp.v3i5.740

[33] Nesti T, Boddana S, Yaman B. Ultra-Sonic Sensor based Object Detection for Autonomous Vehicles. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE; 2023. p. 210–218. 10.1109/cvprw59228.2023.00026

How to cite this article

Al-Saleh AH, Al-Obaidi MAM, Mahdi SM, Al-Zuky AA. A New Procedure for Automatic Vehicle License Plate Recognition Monitoring System via CNN and RCNN and Arduino . Journal of University of Anbar for Pure Science. 2026; 20(1):262-271. doi:10.37652/juaps.2025.161328.1428