Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.12 n.58, p. 475-486. March/April, 2026.
DOI: https://doi.org/10.5935/jetia.v12i58.3138

RESEARCH ARTICLE OPEN ACCESS

A YOLO-BASED AUTOMATIC BANGLADESHI VEHICLE LICENSE PLATE RECOGNITION SYSTEM

Md. Ashraful Islam[1], Naimul Amin[2], Ali Ahmmed[3], Nushrat Jahan Nishita[4]

[1, 2, 3, 4]Department of Electrical and Electronic Engineering, Bangladesh Army International University of Science & Technology (BAIUST), Cumilla-3501, Bangladesh

[1]https://orcid.org/0000-0002-2406-1938, [2]https://orcid.org/0009-0006-9869-809X
[3]https://orcid.org/0009-0001-1247-4073, [4]https://orcid.org/0009-0008-0783-5090

E-mail: ashrafulislam.eece@gmail.com, naimulamin.na@gmail.com, ali726820ahmed@gmail.com, nishitamajumder577@gmail.com

ARTICLE INFO

Article History
Received: December 14, 2025
Reviewed: January 1, 2026
Accepted: January 15, 2026
Published: March 31, 2026

Keywords:
Image processing,
Bangladeshi vehicle,
License plate detection,
Character recognition,
YOLO.

ABSTRACT

In this paper, a YOLO based automatic Bangladeshi vehicle license plate recognition system is proposed. The proposed system has four parts: license plate detection, extracting the region of interest (ROI), applying image processing to the ROI, and character segmentation & recognition. In the detection & extraction stage, the system receives a vehicle image and then detects the license plate and extracts the plate region. The recognition part consists of three consecutive stages: city name recognition, vehicle type recognition and vehicle serial number recognition. As the presented method recognizes the license plate characters in three consecutive stages, a data serialization algorithm is proposed to serial the data. The dataset contains 500 license plate images from four major cities of Bangladesh. The images are used to train, test and validate the proposed model. The proposed method has provided very impressive results and outperformed many other existing methods.

I. INTRODUCTION

Bangladesh is one of the most densely populated countries in the world. According to [1], the population of Bangladesh reaches approximately 171.5 million in 2022. Due to rapid population growth and urbanization, the number of vehicles increases significantly every year [2]. By November 2023, around 5.93 million motor vehicles are registered in Bangladesh. This large number of vehicles in a highly populated country leads to various traffic-related problems, including road accidents, vehicle theft, number plate falsification, unregistered vehicles, overloading, and over-speeding [3]. In addition, many vehicle owners do not renew their vehicle tax-token after expiration, which causes financial loss for the government [3]. To control these issues, traffic management authorities regularly conduct mobile courts and file legal cases against offenders. A large number of cases are filed between January 2023 and June 2023 [3]. However, manual checking of vehicle documents and license plates requires a large amount of time, effort, and manpower. Therefore, monitoring and managing traffic manually becomes difficult, inefficient, and time-consuming. This situation highlights the importance of developing an automated and intelligent traffic monitoring system.

Automatic License Plate Recognition (ALPR) is an important component of modern Intelligent Transportation Systems (ITS) that uses computer vision and machine learning techniques to automatically detect and recognize vehicle license plates [4]. ALPR systems are widely used in applications such as toll collection, parking management, traffic law enforcement, and road traffic monitoring [4–6]. For Bangladesh, such a system can help traffic police quickly identify unregistered vehicles and vehicles with expired tax-tokens by simply scanning the license plate. This improves efficiency, reduces human effort, and supports better law enforcement. Normally, ALPR system consists of three stages named as (i) License plate detection and localization, (ii) Character segmentation, and (iii) Character recognition [4-6]. At the first stage, the license plate is located from images or from a real-time camera system. Different techniques and algorithms like Template matching [7], Morphological operations [8], Histogram Analysis [5], and Machine learning based techniques/models: You only look once (YOLO) [9-11], other neural networks [12], [13] etc are used to detect the license plates from the images.

Then the characters of the detected license plate are segmented. Techniques applied to segment the characters are connected component analysis [5], [7], [8], dynamic programming [14], machine learning based approaches [9-13] etc. At the third stage, each segmented character are classified and recognized. Optical character recognition [15], [16], Template matching [8], [17], Machine learning based approaches/models: You only look once (YOLO) [10], [11], Support vector machine (SVM) [5], other neural networks [7, 9] are widely used to recognize the segmented characters. In Bangladesh, vehicle number plates, commonly known as registration plates, follow specific formats based on vehicle type and are printed using Bengali characters [5]. The Bangladesh Road Transport Authority (BRTA) issues plates in two main categories: private and commercial vehicles. Private vehicles use black text on a white background, while commercial vehicles use black text on a green background, as shown in Figure 1(a) [5]. A Bangladeshi license plate contains two lines of text (Figure 1(b) [8]). The upper line includes the city name, the word “metro,” and the vehicle class, while the lower line contains six digits, where the first two indicate the vehicle class (ranging from 11 to 99) and the last four represent the registration number. The BRTA uses 33 Bangla alphabets to denote vehicle classes-অ, ই, উ, এ, ক, খ, গ, ঘ, ঙ, চ, ছ, জ, ঝ, ট, ঠ, দ, ধ, ত, থ, ড, ঢ, ন, প, ফ, ব, ভ, ম, য, র, ল, শ, স, হ-and Bangla numerals (০-৯), which correspond to the digits 0-9 [5].

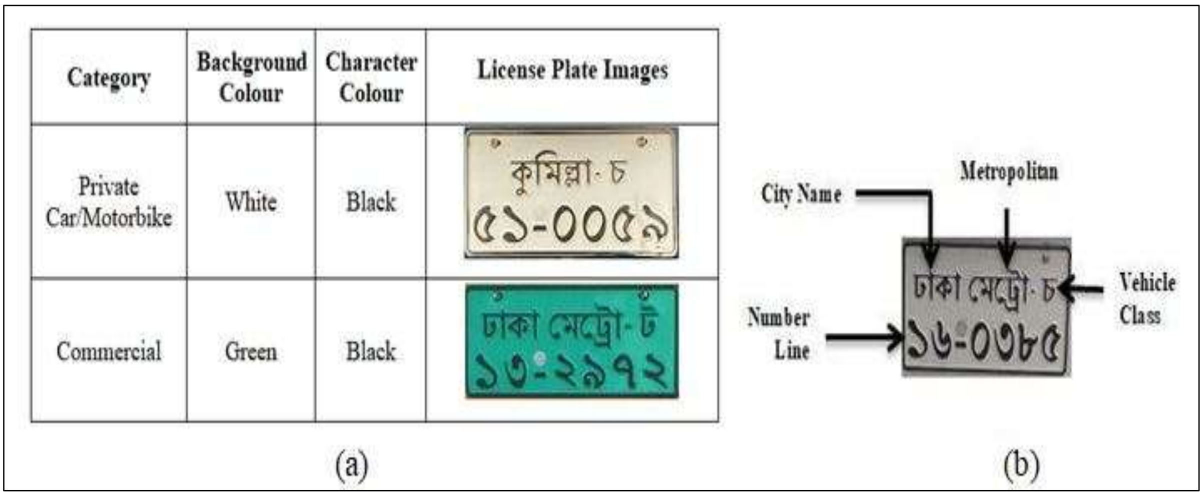

Figure 1: (a) Bengali license plate types according to vehicle category [5], and (b) Bangladeshi license plate representation.
Source: [8].

Despite global progress in ALPR systems, research on Bangladeshi license plate recognition remains limited due to challenges such as complex Bangla characters as mentioned above, low-quality images, different lighting conditions, skewed or tilted plates, and limited public datasets. Several researchers attempt to address these challenges using different approaches. Shahed et al. [17] propose a morphological operation-based system combined with connected component analysis and template matching, achieving 95% accuracy. Other works also use morphological operations, Otsu thresholding, and template matching with small datasets, achieving accuracy between 90% and 95% [18]. Geometric transformation and projection-based segmentation combined with neural networks are applied in [19], resulting in 92.77% accuracy. More recent studies use deep learning methods. Rabbani et al. [20] apply a CNN-based character recognition model after morphological detection and segmentation but achieve only 82.32% accuracy due to limited training data. Chandra et al. [21] combine morphological operations, projection-based segmentation, CNN classifiers, and Optical Character Recognition (OCR), achieving around 95% accuracy. Saif et al. [10] use YOLOv3 models for both license plate detection and character recognition.

Other studies use super-resolution techniques [7], Support Vector Machine (SVM) classifiers [5], and CNN-based models to improve recognition accuracy. However, many existing methods rely on traditional image processing, small datasets, or separate models, which limits robustness and generalization in real-world conditions. Based on existing limitations, this research develops an efficient and robust Automatic License Plate Recognition (ALPR) system for Bangladeshi vehicles using a YOLO-based convolutional neural network. The proposed system works in four main steps: license plate detection, region of interest (ROI) extraction, image enhancement with perspective correction, and character segmentation and recognition. A YOLO-based multiple object detection method is used to segment and recognize Bangla characters, and a data serialization algorithm is applied to place the recognized characters in the correct order. To improve system robustness, effective image enhancement techniques and a newly created Bangladeshi license plate dataset is used. For real-world use, the complete ALPR process is implemented in a user-friendly simulation software developed with Python-Tkinter, making the system suitable for practical applications.

II. PROPOSED METHODOLOGY

In this section, the proposed methodology and all the details of the supporting techniques and algorithms to implement the proposed ALPR system are broadly discussed. The proposed system can efficiently detect and recognize the Bangladeshi license plates. It consists of four major stages: license plate detection, extracting the region of interest (ROI), image processing to prepare it for the next stage, and character segmentation & recognition. The block diagram of the proposed methodology is given in Figure 2.

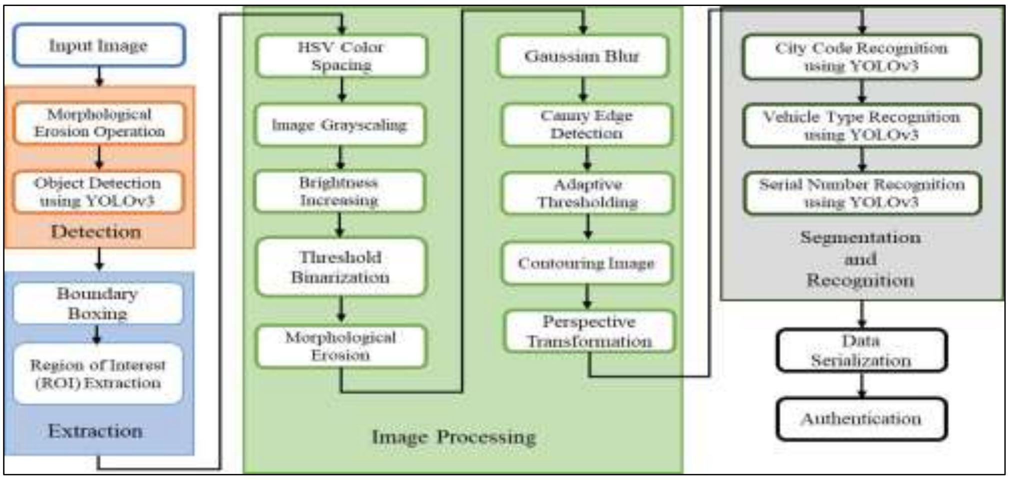

Figure 2: Block diagram of the proposed ALPR system.
Source: Authors, (2026).

II.1 DETECTION AND EXTRACTION OF LICENSE PLATE AREA

The first step to recognize any license plate is detecting and extracting the plate area from the input image. Precise detection of the plate area (ROI) will increase the probability of achieving higher accuracies in the next consecutive stages [22]. The details of the detection and extraction process of the proposed method are described below:

II.1.1 Detection of License Plate

At first, the quality of the input image is improved through morphological erosion operation [23]. This process is done by using the "cv2.erode" function of the python OpenCV framework. The eroded image is then entered into the CNN based license plate detection model. We have used YOLOv3 as the CNN model in our proposed method. Our YOLOv3 based license plate detection model consists of 53 convolutional layers and 5 max pooling layers. We have used the leaky rectified linear unit (leaky ReLU) as the activation function. The value of learning rate, momentum, decay, batch size, and epochs are 0.001, 0.9, 0.0005, 64, and 100, respectively. In our license plate detection model, we have used the same loss function used in [10]. This model returns four-pixel coordinates: x, y, w, h where the number plate area belongs as shown in Figure 3(c). The final output of YOLOv3 model consists of bounding box coordinates, object class probabilities, and confidence scores [10]. The confidence score reflects the confidence that an object is present in a particular bounding box. Figure 3 represents the detection of license plate and extracting the pixel coordinates of the plate area from the input image.

Figure 3: License plate detection: **(a)** input image, (b) eroded image, and (c) pixel coordinates (x, y, w, h).
Source: Authors, (2026).

II.1.2 Extraction of License Plate Area

After getting the desired pixel coordinates, we have applied the "cv2.rectangle" function to draw a boundary box rectangle on the erosion image with specified coordinates and color. After that, the license plate area has been extracted. The extraction of the license plate area consists of two main steps: boundary box generation and region of interest (ROI) extraction. The boundary box generation process begins with the eroded image as input. Using the top-left coordinates (x, y) and the width and height (w, h) of the detected plate area, a rectangular boundary box is drawn around the license plate. The rectangle is rendered in yellow color with RGB values (255, 255, 0) and a line thickness of 2 pixels. This step visually highlights the location of the license plate within the image. After the boundary box is drawn, the license plate area is cropped from the image to create the region of interest (ROI). To ensure that the edges of the rectangle do not include unwanted background, a margin of 10 pixels is applied on all sides. Specifically, the ROI is extracted using the coordinates (y+10):(y+h−10) for the rows and (x+10):(x+w−10) for the columns. The resulting cropped image contains only the license plate, which is then used for subsequent image processing and character recognition. Figure 4 represents some examples of the extracted license plate from the eroded image.

Figure 4: License plate extraction: (a) boundary box drawn images, (b) extracted license plates.
Source: Authors, (2026).

II.2 IMAGE PROCESSING

Figure 4 shows that the extracted ROI may contain unwanted areas and suffer from blurriness or low quality. Therefore, image processing techniques are applied to enhance quality and remove extraneous portions. Additionally, real-time images may have angular or axial deviations, which are corrected before segmentation and recognition. The image processing steps are described below:

II.2.1 Image Gray Scaling and Brightness Increasing

To reduce image complexity, the extracted license plate RGB image is first converted to the HSV color space using the *cv2.cvtColor* function and then split into hue, saturation, and value channels with *cv2.split* to extract the pixel values of the value channel. The HSV image is then merged and converted back to RGB. Using Eq. (1) [8], the RGB image is transformed into a grayscale image. The brightness of the grayscale image is subsequently enhanced by scaling the pixel values of the value channel by 1.5 times. The effect of these operations is shown in Figure 5.

$$Gray\ scale\ image = 0.299 * R + 0.587 * G + 0.114 * B \tag{1}$$

Figure 5: Image gray scaling and brightness increasing: (a) extracted license plate image (RGB image), (b) gray-scale image, and (c) brightness increased gray-scale image.
Source: Authors, (2026).

II.2.2 Threshold Image Binarization and Morphological Erosion Operation

In this step, the brightness-enhanced grayscale image is converted into a binary image using threshold binarization, where each pixel is assigned either black or white, as shown in Figure 6(b). The OpenCV function *cv2.threshold* is applied with a threshold value of 127. Morphological erosion is then performed on the binary image to expand the number plate border pixels, using a 5×5 kernel, as illustrated in Figure 6(c). Finally, Gaussian blur is applied to reduce noise while preserving important image details, using the *cv2.GaussianBlur* function with a 5×5 kernel, as shown in Figure 6(d).

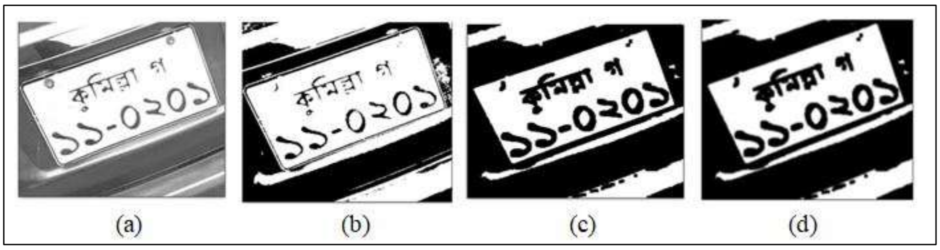

Figure 6: Threshold image binarization and morphological erosion operation: (a) brightness increased gray-scale image, (b) binary image, (c) eroded image, and (d) Gaussian blurred image.
Source: Authors, (2026).

II.2.3 Edge Detection and Adaptive Thresholding

In this step, the edges of the Gaussian blurred image are detected using the canny edge detector. The "*cv2.Canny*" function is used to perform this edge detection. After that, we have applied adaptive thresholding operation to extend the single line edges and to fill up some empty pixels in the boundary region. The effect of these processes is depicted in Figure 7.

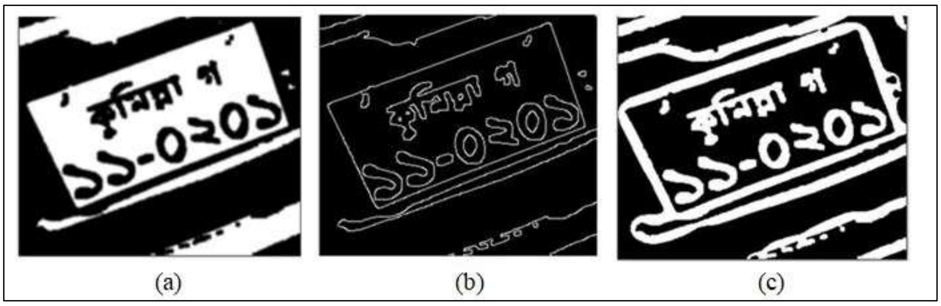

Figure 7: Edge Detection and Adaptive Thresholding: (a) Gaussian blurred image, (b) edge detected image, and (c) image after adaptive thresholding.
Source: Authors, (2026).

II.2.4 Contour Detection and Perspective Transformation

In license plate image segmentation and recognition, object contour detection plays a crucial role. The OpenCV function *cv2.findContours(image, mode, method)* is used to detect contours, where *cv2.RETR_TREE* identifies the hierarchical structure of contours and *cv2.CHAIN_APPROX_NONE* preserves all contour points without approximation. Among the detected contours, the quadrilateral contour with four points is selected, and a boundary box is drawn around it, as shown in Figure 8(b). Real-time images may suffer from angular and axial deviations, which can complicate segmentation and recognition. To address this, perspective transformation is applied to correct skewness and tilt [24]. Using *cv2.getPerspectiveTransform(src, dst)*, a transformation matrix is calculated based on source and destination points, and the image is warped with *cv2.warpPerspective* to achieve the desired perspective, as shown in Figure 8(c). This process also standardizes the output image to a fixed size of 400×250 pixels for subsequent character segmentation and recognition.

Figure 8: Contour detection and perspective transformation: (a) image after adaptive thresholding, (b) contoured image, and (c) image after perspective transformation.
Source: Authors, (2026).

II.3 CHARACTER SEGMENTATION AND RECOGNITION

Character segmentation & recognition process is one of the major parts of an ALPR system. In this stage, the characters of the license plate are segmented and then recognized using various techniques. Here, to segment & recognize the characters, we have used the same type of YOLOv3 model and hyper-parameters that we used for the detection purpose. The details of the character segmentation & recognition process of the proposed method are described below:

City name recognition is the very first step of our series machine learning approach. In our research, we have worked with four cities out of which three are metropolitan areas. So, we have used five classes for city name recognizer. Table 1 represents the city codes, their meaning and the trained class names.

Table 1: City codes, their meaning and the trained class names for City name recognizer.

City Code	Meaning in Bangla and in English	Trained Class Name
ঢাকা	ঢাকা (Dhaka)	dhaka
চট্ট	চট্টগ্রাম (Chattogram)	chatto
কুমিল্লা	কুমিল্লা (Comilla)	comilla
সিলেট	সিলেট (Sylhet)	sylhet
মেট্রো	মেট্রোপলিটন (Metropolitan)	metro

Source: Authors, (2026).

Vehicle type recognition is the second step of our series machine learning approach. After segmenting and recognizing the city name (with metro), the bounding box surrounded image of the previous step enters into the vehicle type recognizer. In this research, we have worked on six types of vehicles. The vehicle types, their meaning and the trained class names are shown in Table 2.

Table 2: Vehicle types, their meaning and the trained class names for Vehicle type recognizer.

Vehicle Type	Meaning	Trained Class Name
হ	Motorbike (80cc-125cc)	ha
গ	Private Car (1500cc-1800cc)	ga
ল	Motorbike (135cc-200cc)	la
চ	Microbus	cha
ঘ	Jeep	gha
ঝ	Bus or Coaster	jha

Source: Authors, (2026).

The final stage of the series machine learning recognition process is vehicle serial number recognition with ten classes. The classes are ten Bengali number digits (০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯) which is equivalent to 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. City name, vehicle type, and vehicle serial number recognition are performed using a multi-class detection approach based on the YOLOv3 convolutional neural network. For each recognizer, the model detects the desired class within the image, returns the pixel coordinates, and draws a rectangular bounding box around the detected region, labeling it with the corresponding class name, as shown in Figures 9(b)–9(d). Each recognition task was trained independently using the same YOLOv3 architecture and hyper-parameters, while differing only in class definitions and annotations.

Figure 9: Character Segmentation & Recognition process: (a) image after perspective transformation, (b) city name recognition to text, (c) vehicle type recognition to text and, (d) serial number recognition to text.

Source: Authors, (2026).

II.4 DATA SERIALIZATION FOR THE RECOGNIZED CHARACTERS

The series machine learning method may recognize license plate characters in a random order. For example, a plate containing both “dhaka” and “metro” may detect “metro” before “dhaka”. To address this issue, a data serialization algorithm based on the smallest x-axis coordinate is proposed to arrange the recognized characters in the correct sequence. The algorithm of the data serialization process is given below:

Algorithm 1: Data Serialization of Recognized Characters.

Input: Bounding boxes with class IDs and confidences
Output: Serialized text string
1: *Apply Non-Maximum Suppression (NMS) to city code and vehicle type bounding boxes and extract their labels.*
2: *Apply NMS to all remaining bounding boxes.*
3: *Sort the selected bounding boxes in ascending order based on their x-axis coordinates.*
4: *Append the corresponding class labels sequentially from left to right to form the final serialized text.*

Source: Authors, (2026).

The serialization process is divided into two steps. In the first step, the detected city name and vehicle type bounding boxes are filtered using Non-Maximum Suppression (NMS), and their corresponding labels are extracted. In the second step, the bounding boxes of the vehicle serial number are also filtered using NMS and then sorted in ascending order based on their x-axis coordinates. The labels are concatenated sequentially according to this order, ensuring that characters with smaller x-axis values appear first. This two-step process ensures proper left-to-right serialization of all recognized license plate components.

II.5 DEVELOPMENT OF SIMULATION SOFTWARE

To integrate the proposed system with a smart city framework, a Tkinter-based software application was developed to implement the complete license plate recognition process. The user-friendly interface allows users to select an input image using the *Open* button and execute the recognition process using the *Detect* button. Intermediate output images are displayed sequentially, and the recognized license plate number is shown at the top of the interface. For authentication, a local server-based database stores vehicle plate numbers and related information in an Excel file. The software matches the recognized plate number with the database and displays the corresponding vehicle information in the interface. This integrated application provides a comprehensive solution for deploying ALPR systems in smart city environments. The user interface of the developed simulation software is shown in Figure 10.

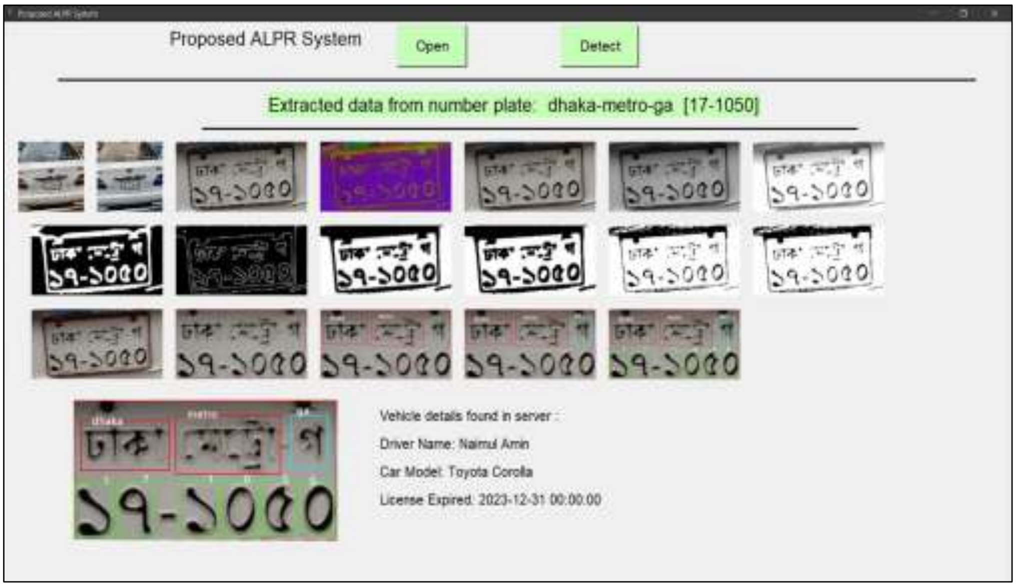

Figure 10: User interface of the developed simulation software.
Source: Authors, (2026).

III. EXPERIMENTAL SETUP

A standard and sufficiently large dataset is essential for object & pattern recognition research, as CNNs perform better with more training data. For this study, a dataset of 500 Bangladeshi vehicle license plate images was prepared. Among them, 400 images were captured in real time from different locations in Dhaka, Comilla, Sylhet, and Chittagong using a 20-megapixel mobile phone camera, while the remaining images were collected from [25] and [26]. All images were resized to 500×400 pixels and stored in JPEG format. The dataset includes diverse conditions such as daylight and low-light scenes, motion blur, varying backgrounds, weather conditions, and skewed or tilted views. To reduce overfitting, data augmentation was manually applied to each image by randomly performing two operations-rotation, contrast adjustment, Gaussian noise addition, and HSV-based saturation and hue modification [27]. As a result, each original image generated two additional samples, making the dataset more robust. Table 3 depicts the summary of the images collected for this research and Figure 11 represents some of the original and augmented images from the license plate image dataset.

Table 3: Summary of license plate image dataset.

Date	**Location (in Bangladesh)**	**No. of images**	
		Before augmentation	**After augmentation**
04 October, 2023	Kandirpar, Comilla	50	150
09 October, 2023	Cantonment Bus Stand, Comilla	60	180
12 October, 2023	Mohakhali Bus Stand, Dhaka	80	240
13 October, 2023	Banashri, Dhaka	80	240
16 October, 2023	Kadamtoli Bus Stand, Sylhet	50	150
20 October, 2023	Dampara Bus Stand, Chittagong	80	240
30 October, 2022	Collected from [25]	50	150
05 November, 2022	Collected from [26]	50	150
	Total	**500**	**1500**

Source: Authors, (2026).

Figure 11: Sample images from license plate image dataset.
Source: Authors, (2026).

In the proposed method, for detection purpose, we have split the image dataset to 60%, 20% and 20% for training, validating and testing the models. We have used 900 images for training, 300 images for validating and 300 images for testing the proposed ALPR detection model. Although the original images remain the same, separate YOLO-formatted annotation sets were created for each recognition stage. For city name recognition, bounding boxes were annotated for five classes (dhaka, chatto, comilla, sylhet, and metro). For vehicle type recognition, cropped outputs from the city name detection stage were annotated with six vehicle-type classes. Finally, for vehicle serial number recognition, individual Bengali digits (০-৯) were annotated as ten distinct classes, with each digit labeled by a separate bounding box. The computing environment of our research is a laptop having an operating system of 64 bit Windows 10 Pro, Intel Core i7 CPU, 1TB hard disk, SSD of 500GB, and 8 GB RAM. We have used a GPU of Gigabyte RX 6900 XT with 16 GB RAM to execute the processes faster.

IV. RESULTS AND DISCUSSIONS

In this section, we are going to discuss about the training, validation and testing results of the proposed ALPR method. We will also compare the achieved results with the other proposed methods.

IV.1 TRAINING AND VALIDATION RESULTS

The YOLOv3 based CNN models for the license plate area detection, and the character recognition is trained, and validated with 1200 vehicle license plate images. At first, we have trained the models with the images and then validated the models. Our proposed systems have achieved outstanding training and validation results as depicted in Table 4. The average accuracies of the license plate detection, city name, vehicle type and serial number recognition models in training & validation are 97.24%, 96.94%, 95.8%, and 96.05% respectively.

Table 4: Training and validation accuracy table for YOLOv3 models.

Stages	Phases	Sample Size	Accuracy
Detection	Training	900	97.5%
	Validation	300	96.97%
City Name Recognition	Training	900	97%
	Validation	300	96.88%
Vehicle Type Recognition	Training	900	96.2%
	Validation	300	95.4%
Serial Number Recognition	Training	900	96.5%
	Validation	300	95.6%

Source: Authors, (2026).

IV.2 TESTING RESULTS

The detection stage of the proposed ALPR method is tested with 300 license plate images. While testing the model, the proposed method has showed very robust performance in different challenging scenarios. The confusion matrix of the license plate detection model is presented in Table 5. The proposed model has successfully detected 291 license plates out of 300 vehicle images. So, the accuracy of the proposed license plate detection model is 97% as calculated using Eq. (2).

Table 5: Confusion matrix for license plate detection model

		Predicted	
Actual		Positive	Negative
	Positive	TP[a]= 291	FN[a]= 007
	Negative	FP[a]= 002	TN[a]= 000
	[a]TP= True Positive, TN=True Negative, FP= False Positives and FN= False Negative		

Source: Authors, (2026).

$$Accuracy\ of\ a\ model = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2)$$

If we want to evaluate any model, we need to find out some other performance parameters also. Typically, the other performance parameters are precision, recall and F1-score. Eq. (3-5) represents the equations of these parameters.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision+Recall} \times 100\% \quad (5)$$

The model has provided an impressive result of F1-score of 98.48% along with precision of 99.32% and recall of 97.65% as shown in Table 6.

Table 6: Performance parameter values for license plate detection model.

Stages	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Detection	99.32	97.65	98.48

Source: Authors, (2026).

In the extraction stage, our proposed method has successfully extracted 279 license plates out of 291 by providing an accuracy of 95.88%. We have used Eq. (6) to find out the accuracy.

$$Accuracy = \frac{No.of\ license\ plates\ successfully\ extracted}{No\ of\ detected\ images} \times 100\% \quad (6)$$

The recognition phase represents the final stage of the proposed ALPR system, where extensive testing was conducted. The city name, vehicle type, and serial number recognition models were evaluated using 279 extracted license plate images, comprising 279 city name characters, 158 *metro* characters, 279 vehicle type characters, and 1,674 Bengali digits for vehicle serial numbers. The corresponding confusion matrices are presented in Tables 7-9. Overall, the recognition models demonstrate high performance, with only a few misclassifications. The system correctly recognized 423 out of 437 city names, 264 out of 279 vehicle types, and 1,593 out of 1,674 digits, achieving accuracies of 96.79%, 94.62%, and 95.16%, respectively. Eq. (2) is used to measure these accuracies.

Table 7: Confusion matrix for city name recognition model.

		Predicted				
	City Name	**ঢাকা (dhaka)**	**চট্ট (chatto)**	**কুমিল্লা (comilla)**	**সিলেট (sylhet)**	**মেট্রো (metro)**
Actual	**ঢাকা (dhaka)**	101	001	000	000	001
	চট্ট (chatto)	001	051	000	000	003
	কুমিল্লা (comilla)	000	000	066	001	000
	সিলেট (sylhet)	000	000	002	052	000
	মেট্রো (metro)	000	004	000	000	153

Source: Authors, (2026).

Table 8: Confusion matrix for vehicle type recognition model.

		Predicted					
	Vehicle Type	**হ (ha)**	**গ (ga)**	**ল (la)**	**চ (cha)**	**ঘ (gha)**	**ঝ (jha)**
Actual	**হ (ha)**	19	01	00	00	01	00
	গ (ga)	00	83	00	00	02	00
	ল (la)	00	01	23	00	00	00
	চ (cha)	00	01	00	32	01	00
	ঘ (gha)	00	01	00	00	28	02
	ঝ (jha)	00	01	00	00	03	79

Source: Authors, (2026).

Table 9: Confusion matrix for serial number recognition model.

		Predicted									
	Digits	**০ (0)**	**১ (1)**	**২ (2)**	**৩ (3)**	**৪ (4)**	**৫ (5)**	**৬ (6)**	**৭ (7)**	**৮ (8)**	**৯ (9)**
Actual	**০ (0)**	197	002	000	002	004	002	000	002	000	000
	১ (1)	001	271	001	001	001	000	000	000	000	004
	২ (2)	000	005	193	002	000	000	000	000	000	002
	৩ (3)	004	000	000	178	001	000	000	001	000	002
	৪ (4)	003	001	000	000	123	000	000	001	001	000
	৫ (5)	003	000	000	001	000	166	005	000	001	000
	৬ (6)	002	000	000	001	000	004	131	000	001	000
	৭ (7)	004	001	000	001	001	000	000	126	000	000
	৮ (8)	001	000	000	000	002	000	000	001	113	000
	৯ (9)	001	003	000	001	000	000	000	000	000	95

Source: Authors, (2026).

From the confusion matrix of the above tables, it is observed that the city name recognition model faced some difficulties while differentiating “চট্ট” and “মেট্রো”. In some cases, the model has predicted “মেট্রো” as “চট্ট” and vice versa. The vehicle type recognition model has predicted “গ” as “ঘ”, “ঝ” as “ঘ” and vice versa a few times. Vehicle serial number recognition model has also faced some difficulties while recognizing the digits. As license plate images contain noises and dusts, sometimes a digit is looked like another digit such as “০” as “১”or “৪”, “১” as “৯”, “২” as “১”, “৩” as “০”, “৪” as “০”, “৫” as “৬” or “০”, “৬” as “৫” or “০”, “৭” as “০”, “৯” as “১”. The proposed model has provided wrong predictions in some of the above cases. The performance parameters of the license plate character recognizer models are given in Table 10.

Table 10: Performance parameter values for license plate character recognition models.

Stages	**Precision (%)**	**Recall (%)**	**F1-score (%)**
City Name	99.76	97.02	98.37
Vehicle type	99.62	94.96	97.23
Bengali digits	99.75	95.39	97.52

Source: Authors, (2026).

We know that, when the class distribution is uneven, it is better to calculate the F1-score than the accuracy [5]. As in our proposed system, the number of false positives and false negatives isn’t near to each other; the F1-score better represents the state of the models than the accuracy calculation. It can be observed from Table 10 that, our proposed character recognition models have provided outstanding F1-scores of 98.37%, 97.23%, and 97.52% for city name recognizer, vehicle type recognizer and serial number recognizer respectively. The Table 11 depicts the overall summary of the results for the proposed method.

Table 11: Results of different stages for the proposed method.

Stages	Sample Size	Passed	Accuracy	F1-score
Detection	300	291	97%	98.48%
Extraction	291	279	95.88%	---
City Name Recognition (including ‘metro’)	437	423	96.79%	98.37%
Vehicle Type Recognition	279	264	94.62%	97.23%
Serial Number Recognition	1674	1593	95.16%	97.52%
Average accuracy, and F1 score of the proposed method			95.89%	97.90%
Average recognition accuracy and F1 score of the proposed method			95.52%	97.71%

Source: Authors, (2026).

IV.3 COMPARING THE PROPOSED METHOD WITH OTHER EXISTING METHODS

A comparison between the proposed method and existing Bangladeshi ALPR systems is presented in Table 12. Previous studies have aimed to improve robustness, usability, and recognition accuracy at different stages of ALPR. Ashrafee et al. [25] proposed a MobileNet SSDv2 and Google Vision API-based system with a relatively large dataset; however, their approach achieved limited accuracy, did not employ image quality enhancement techniques, and suffered from slow inference speed. The datasets used in [5] and [7] were comparatively small; notably, Alam et al. [7] trained their CNN-based character recognizer using only 500 images without applying data augmentation, increasing the risk of overfitting. Moreover, the methods in [4], [5], and [7] did not incorporate geometric correction techniques, which may limit their ability to handle angular or axial deviations, while the system in [4] failed to detect license plates in the presence of irrelevant objects. In contrast, the proposed method addresses these limitations by incorporating image enhancement, geometric correction, and data augmentation, resulting in improved recognition accuracy and robustness, as shown in Table 12. Additionally, the overall processing time of the proposed system is lower than that of existing methods. To further enhance usability, a dedicated simulation software was developed, which was not reported in previous works. These advantages demonstrate that the proposed method outperforms existing Bangladeshi ALPR systems.

Table 12: Comparison of proposed method with other existing methods.

References	**Methods**	**Sample Size**	**Testing Accuracy**	**Processing Time**
Proposed Method	CNN (YOLOv3), Morphological Image Processing, Geometrical Transformation	1500	Detection: 97% F1-score: 98.48% Extraction: 95.88% Recognition: 95.52% F1-score: 97.71%	108 ms[b]
[25]	Mobile Net SSDv2, Google Vision API	1000	Detection: 82.7% Recognition: 60.8%	---
[7]	Template Matching, CNN	700	Detection: 98.2% Recognition: 90.9%	111 ms[b]
[5]	Morphological Operation, Histogram Analysis, SVM	630	Detection: 91% Character Extraction: 94.6% Recognition: 100%	---
[4]	CNN (YOLOv3), Tesseract	---	Detection: 95% Recognition: 91.38%	---
	[b]ms= milliseconds			

Source: Authors, (2026).

V. CONCLUSIONS

This paper presents a YOLOv3-based ALPR system for detecting, extracting, and recognizing Bangladeshi vehicle license plates. A primary dataset of 500 vehicle images was prepared, covering diverse conditions such as varying backgrounds, weather, illumination, low-light scenarios, and skewed or tilted views. To address the data requirements of CNNs and reduce overfitting, manual data augmentation was applied, increasing the dataset size to 1,500 images. The models were trained, validated, and tested using 900, 300, and 300 images, respectively. Perspective transformation was employed to correct angular and axial distortions, and a series machine learning approach was introduced for character recognition, along with a data serialization algorithm to preserve character order. Extensive testing demonstrated strong performance, achieving accuracies of 97% for license plate detection, 95.88% for extraction, and 95.52% for character recognition. Additionally, a simulation software was developed to enhance system usability for traffic management authorities. Future work will focus on expanding the dataset with more diverse samples to further improve performance and support additional vehicle classes.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Methodology: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Investigation: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Discussion of results: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Writing – Original Draft: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Writing – Review and Editing: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Resources: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Supervision: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.
Approval of the final text: Md. Ashraful Islam, Naimul Amin, Ali Ahmmed, Nushrat Jahan Nishita.

VII. REFERENCES

[1] United Nations Statistics Division (UNSD), "UNdata on population, surface area and density," 2022. [Online]. Available: https://data.un.org/_Docs/SYB/PDFs/SYB66_1_202310_Population,%20Surface%20Area%20and%20Density.pdf

[2] Bangladesh Road Transport Authority (BRTA), "Report on number of registered vehicles in whole Bangladesh," 2023. [Online]. Available: http://www.brta.gov.bd/site/page/74b2a5c3-60cb-4d3c-a699-e2988fed84b2/Number-of-registered-Vehicles-in-Whole-BD

[3] Bangladesh Road Transport Authority (BRTA), *Annual Report for the Fiscal Year 2022–23*, Dhaka, Bangladesh, 2023. [Online]. Available: http://www.brta.gov.bd/sites/default/files/files/brta.portal.gov.bd/annual_reports/9c5532fd_05da_4e14_9feb_d606ce97bbbf/2023-11-01-08-44-d71a318c9e1f579dbb115d83edbeb110.pdf

[4] M. N. I. Suvon, R. Khan, and M. Ferdous, "Real time Bangla number plate recognition using computer vision and convolutional neural network," in *Proc. IEEE 2nd Int. Conf. Artificial Intelligence in Engineering and Technology (IICAIET)*, Kota Kinabalu, Malaysia, 2020, pp. 1–6, doi: 10.1109/IICAIET49801.2020.9257843.

[5] R. Islam, M. R. Islam, and K. H. Talukder, "An efficient method for extraction and recognition of Bangla characters from vehicle license plates," *Multimedia Tools and Applications*, vol. 79, pp. 20107–20132, 2020, doi: 10.1007/s11042-020-08629-8.

[6] R. Laroca *et al.*, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *IET Intelligent Transport Systems*, vol. 15, pp. 483–503, 2021, doi: 10.1049/itr2.12030.

[7] N.-A. Alam, M. Ahsan, M. A. Based, and J. Haider, "Intelligent system for vehicles number plate detection and recognition using convolutional neural networks," *Technologies*, vol. 9, no. 1, p. 9, 2021, doi: 10.3390/technologies9010009.

[8] M. A. Islam and A. Habib, "An effective method for the recognition and verification of Bangladeshi vehicle digital number plates," *Journal of Computer Science*, vol. 17, no. 11, pp. 1059–1070, 2021, doi: 10.3844/jcssp.2021.1059.1070.

[9] S. Abdullah, M. M. Hasan, and S. M. S. Islam, "YOLO-based three-stage network for Bangla license plate recognition in Dhaka metropolitan city," in *Proc. Int. Conf. Bangla Speech and Language Processing (ICBSLP)*, Sylhet, Bangladesh, 2018, pp. 1–6, doi: 10.1109/ICBSLP.2018.8554668.

[10] N. Saif *et al.*, "Automatic license plate recognition system for Bangla license plates using convolutional neural network," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2019, pp. 925–930, doi: 10.1109/TENCON.2019.8929280.

[11] M. M. Sarif *et al.*, "Deep learning-based Bangladeshi license plate recognition system," in *Proc. 4th Int. Symp. Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Istanbul, Turkey, 2020, pp. 1–6, doi: 10.1109/ISMSIT50672.2020.9254748.

[12] S. A. Hossain, M. Z. Hassan, and M. A. Masba, "Automatic license plate recognition system for Bangladeshi vehicles using deep neural network," in *Lecture Notes on Data Engineering and Communications Technologies*, Springer, 2021, pp. 91–102, doi: 10.1007/978-981-16-6636-0_8.

[13] T. Islam and R. I. Rasel, "Real-time Bangla license plate recognition system using Faster R-CNN and SSD," in *Proc. IEEE Int. Conf. Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON)*, Dhaka, Bangladesh, 2019, pp. 108–111, doi: 10.1109/RAAICON48939.2019.45.

[14] E. N. Vesnin and V. A. Tsarev, "Segmentation of images of license plates," *Pattern Recognition and Image Analysis*, vol. 16, pp. 108–110, 2006, doi: 10.1134/S1054661806010342.

[15] S. U. Ahmed, F. B. F. Maisha, and M. Hossam-E-Haider, "Bangla license plate detection and recognition system with YOLOv7 and improved custom OCR engine," in *Proc. 4th Int. Conf. Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, Mandya, India, 2022, p. 17, doi: 10.1109/ICERECT56837.2022.10060446.

[16] M. S. H. Onim *et al.*, "Traffic surveillance using vehicle license plate detection and recognition in Bangladesh," in *Proc. 11th Int. Conf. Electrical and Computer Engineering (ICECE)*, Dhaka, Bangladesh, 2020, pp. 121–124, doi: 10.1109/ICECE51571.2020.9393109.

[17] M. T. Shahed *et al.*, "Automatic Bengali number plate reader," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2017, pp. 1364–1368, doi: 10.1109/TENCON.2017.8228070.

[18] M. J. Hossain, M. H. Uzzaman, and A. S. Saif, "Bangla digital number plate recognition using template matching for higher accuracy and less time complexity," *International Journal of Computer Applications*, vol. 181, no. 29, pp. 15–21, 2018, doi: 10.5120/ijca2018918140.

[19] M. K. Hossen *et al.*, "License plate detection and recognition system based on morphological approach and feed-forward neural network," *International Journal of Computer Science and Network Security*, vol. 18, no. 5, pp. 36–45, 2018.

[20] G. Rabbani *et al.*, "Bangladeshi license plate detection and recognition with morphological operation and convolution neural network," in *Proc. 21st Int. Conf. Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2018, pp. 1–5, doi: 10.1109/ICCITECHN.2018.8631937.

[21] S. Chandra *et al.*, "An automated system to detect and recognize vehicle license plates of Bangladesh," in *Proc. 20th Int. Conf. Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2017, pp. 1–6, doi: 10.1109/ICCITECHN.2017.8281781.

[22] S. Sotheeswaran and A. Ramanan, "A coarse-to-fine strategy for vehicle logo recognition from frontal-view car images," *Pattern Recognition and Image Analysis*, vol. 28, pp. 142–154, 2018, doi: 10.1134/S1054661818010170.

[23] K. A. M. Said and A. B. Jambek, "Analysis of image processing using morphological erosion and dilation," *Journal of Physics: Conference Series*, vol. 2071, p. 012033, 2021, doi: 10.1088/1742-6596/2071/1/012033.

[24] K. Wang *et al.*, "Perspective transformation data augmentation for object detection," *IEEE Access*, vol. 8, pp. 4935–4943, 2020, doi: 10.1109/ACCESS.2019.2962572.

[25] A. Ashrafee *et al.*, "Real-time Bangla license plate recognition system for low resource video-based applications," in *Proc. IEEE/CVF Winter Conf. Applications of Computer Vision Workshops (WACVW)*, Waikoloa, USA, 2022, pp. 479–488, doi: 10.1109/WACVW54805.2022.00054.

[26] M. A. Islam, G. M. Chowdhury, and M. N. M. Haque, "Bangla license plate detection, recognition and authentication with morphological process and template matching," in *Proc. 2nd Int. Conf. Emerging Technology (INCET)*, Belagavi, India, 2021, pp. 1–6, doi: 10.1109/INCET51464.2021.9456145.

[27] D. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 2004, doi: 10.1021/ci0342472.