Advanced Techniques for Improved Bangladeshi Number Plate Detection and Character Recognition in Automated Parking Systems

Niaz Ashraf Khan, Md. Ferdous Bin Hafiz
Department of Computer Science and Engineering, University of Liberal Arts Bangladesh, Dhaka, Bangladesh

Article Info

Article history:

Received Mar 4, 2024
Revised Jun 28, 2024
Accepted Jun 30, 2024

Keywords:

Image processing
Number plate
Arduino
Smart parking
MySQL

ABSTRACT

This paper presents a novel technique for efficient extraction of vehicle number plates from camera-captured images and accurate recognition of Bangla characters embedded within them. With the exponential growth of vehicular traffic in densely populated regions like Bangladesh, automation becomes crucial, making vehicle plate recognition pivotal for tracking stolen vehicles and enhancing traffic control measures. Leveraging conventional computer vision and image processing techniques, our proposed system incorporates specific features inherent to Bangladeshi number plates, thus enhancing recognition accuracy. Our application makes use of the OpenCV library to underscore the strength of the algorithm, which has been confirmed through real-time testing across different weather conditions and varying image qualities. The results show a remarkable accuracy rate of 92.3%, affirming our technique's reliability in vehicle number plate detection and character recognition. Moreover, the integration with MySQL database and Arduino UNO enables real-time application in automated parking systems, offering seamless entry procedures and accurate billing, thus addressing critical concerns in modern transportation management systems. Our algorithm not only enhances security measures but also streamlines parking facility management, contributing to safer and more efficient urban mobility solutions.

Corresponding Author:

Niaz Ashraf Khan,
Department of Computer Science and Engineering,
University of Liberal Arts Bangladesh,
Dhaka, Bangladesh
niaz.ashraf@ulab.edu.bd

1. INTRODUCTION

Bangladesh, renowned for its dense population, faces a simultaneous rise in vehicular traffic due to exponential population growth. This surge necessitates automation, where vehicle plate recognition emerges as a pivotal solution. Beyond mere automation, plate detection serves to uncover fraudulent or unregistered number plates, benefiting organizations, universities, and residential complexes with controlled parking systems. However, the complexity of Bangla characters—comprising 11 vowels, 49 consonants, and compound glyphs—poses a unique challenge.

Globally, number plate recognition systems have been extensively researched and implemented to enhance traffic management, security, and automation in parking systems. These systems have been adapted to various languages and regions, each presenting unique challenges. For instance, in countries with simpler character sets, such as English, the recognition process can be relatively straightforward. However, regions with complex scripts, such as Chinese or Arabic, face challenges similar to those in Bangladesh, requiring specialized algorithms for accurate recognition.

This paper delves into a framework capable of accurately detecting and recognizing vehicle number plate characters. The framework entails two primary phases: first, the extraction of the number plate from an image, followed by character segmentation and recognition. The recognized characters are then converted into a string, which undergoes verification against a database. Successful validation prompts the system to signal gate opening for authorized vehicle entry, which later could be used for billing or other secuirity purpose. The database, integral to this process, records vital information including entry and exit times, and overall parking lot occupancy.

Furthermore, adherence to regulations set by the Bangladesh Road Transport Authority (BRTA) underscores the algorithm's design, ensuring compatibility with mandated vehicle plates. This paper provides an approach tailored specifically for the Bangladeshi context, highlighting its critical role in modern parking management systems. The following are the key contributions of our research:

- Tailored algorithmic framework specifically designed for the unique challenges posed by Bangladeshi vehicle number plates.
- Adherence to strict regulations set by the BRTA, ensuring compliance with the required standards.
- Utilization of machine learning techniques for enhanced character recognition and database management.
- Integration with Arduino UNO, enabling fast processing and database integration in real-time parking management scenarios.

In our review of existing literature on number plate recognition, we find notable studies that contribute to the field. Notably, reference [1] presents an approach utilizing image processing techniques similar to ours, although their algorithm hinges on detecting the keyword 'মেট্রো' in number plates, a feature not universal to all vehicle registrations in Bangladesh.

Additionally, [2] showcases MATLAB's Image Processing Toolbox efficacy in reducing recognition errors for Lebanese plates. Furthermore, [3] introduces a novel speed detection framework using Raspberry Pi and cameras for number plate detection, emphasizing cost-effectiveness and simplicity.In Paper [4], a four-stage technique for automated number plate detection and identification (ALPDR) is proposed, leveraging deep learning principles such as Convolutional Neural Networks (CNN), MobileNet, Inception V3, and ResNet 50. This study aims to enhance the efficiency of ALPR systems for applications like toll payment, parking management, and police surveillance.

The authors in paper [5] presents an efficient Vehicle Number Recognition (VNR) system using advanced computer vision techniques. It accurately identifies vehicle identifiers in real-time video streams, employing deep learning for localization and character partitioning on number plates. In Paper [6], an Automatic License Plate Recognition (ALPR) system is created for Saudi car plates using 50 images. Preprocessing, edge detection, noise reduction, and segmentation are performed, followed by OCR for English and Arabic characters. The results achieve 92.4% accuracy for Arabic and 96% for English characters. The authors in paper [7] introduces a CNN-based approach for Indian Number Plate Recognition (INPR), achieving a remarkable 95% accuracy after fine-tuning with real data. By generating synthetic data conforming to Indian standards and employing post-processing techniques specific to Indian number plates, the proposed lightweight model demonstrates practical deployment potential on edge devices like Nvidia Jetson Nano and Intel Neural Compute Stick.

In paper [8], the authors present a robust method for vehicle number plate recognition using mathematical morphological operations, including erosion and dilation. The system effectively detects and translates number plate information from digitally captured images, employing techniques such as image improvement, grayscale transformation, edge detection, and template matching with OCR, resulting in accurate and efficient number number detection. Paper [9] presents an efficient automatic number plate detection and recognition system, utilizing methods such as thresholding, morphological techniques, contour detection, and KNN classification, demonstrating improved accuracy and recognition rates. Paper [10] introduces an automatic number plate detection method leveraging TensorFlow for machine learning, aiming to track vehicles efficiently amid a surge in vehicle numbers and related illegal activities. By employing techniques like image processing, grayscale conversion, and optical character recognition, the system achieves accurate text extraction from vehicle images, facilitating future use for authorities.

Paper [11] introduces the Egyptian Automatic License Plate Recognition (EALPR) dataset, addressing the lack of datasets for Arabic language number plate recognition systems. With 2,450 vehicle images collected from social-media platforms, this dataset enhances the training and evaluation of deep learning models for number plate recognition. Paper [12] presents an automatic number plate recognition (LPR) system tailored for three distinct Iraqi car number plate styles. Using MATLAB R2014a, the algorithm effectively segments and recognizes characters, achieving accurate identification of sample images. Paper [13] proposes an automatic number plate detection and recognition system utilizing a modified U-Net deep learning model. This system addresses challenges in real-time number plate recognition in urban environments, contributing to enhanced traffic management and security measures.

In light of the extensive literature on automated license plate recognition (ALPR) systems, our paper makes several significant technical contributions tailored specifically for the Bangladeshi context. Firstly, we have developed a customized algorithmic framework designed to address the unique challenges posed by Bangladeshi vehicle number plates, encompassing advanced image processing techniques, character segmentation, and recognition algorithms optimized for accurate detection and recognition of Bangla characters, including compound glyphs.

Current systems often struggle with the complexity of Bangla characters, varied lighting conditions, and the diverse range of plate designs. Our paper aims to address these challenges by employing advanced image processing techniques and machine learning algorithms specifically tailored for Bangladeshi number plates.

Additionally, our algorithm design ensures strict adherence to regulations set by the Bangladesh Road Transport Authority (BRTA), guaranteeing compatibility with mandated vehicle plate standards. Leveraging machine learning techniques to enhance character recognition accuracy, while integration with Arduino UNO enables fast processing and seamless database integration in real-time parking management scenarios. Overall, our research significantly advances the field of ALPR technology in Bangladesh, offering a robust and reliable solution for modern parking management systems in the region.

The structure of the remainder of this paper is as follows: Section 2 delves into the Theory behind the proposed algorithm, explaining the foundational principles from image processing techniques to machine learning algorithms. Section 3 presents the proposed methodology, detailing the step-by-step process of plate extraction, color determination, segmentation, character recognition, and integration with Arduino microcontrollers and databases. Following that, Section 4 discusses the Experiment and Results, outlining the testing methodology, success rates, and performance comparisons with other techniques. Finally, Section 5 concludes the paper by summarizing the findings and highlighting the significance of the proposed algorithm in addressing challenges in modern transportation management systems.

2. THEORY

To deploy the algorithm outlined in this paper, we incorporated a variety of methods from traditional image processing, computer vision, and machine learning. Below, we summarize some of the fundamental principles guiding our implementation.

2.1. Grayscale Conversion

Converting an image into binary format is a crucial step in image processing. Prior to this conversion, the image must first be transformed into grayscale. Grayscale images are generated from variations in light intensity or brightness at each pixel. Typically encoded in an 8-bit integer, grayscale intensity offers 256 potential shades of gray, spanning from black (0) to white (255). Grayscale images are easier to process as they reduce computational load and focus on variations in light intensity, making edge detection and subsequent image processing tasks more efficient. The formula utilized for grayscale conversion is:

$$Y = 0.21R + 0.72G + 0.07B \tag{1}$$

Thresholding is a widely used method to reduce noise by removing pixels with extremely low or high values. In our method, we employed Otsu's thresholding technique [14] to convert the grayscale image into a binary format, effectively enhancing its clarity and reducing background noise. An important benefit of Otsu's thresholding method is its capacity to automatically ascertain the optimal threshold without the need for manual intervention. This feature proves invaluable in situations where determining the ideal threshold can be challenging due to variations across different images or when manual adjustment is not feasible.

2.2. Morphological Operations

Morphological operations are essential in image processing, aiding in tasks such as noise reduction and character separation or joining. Our implementation relies on two key morphological techniques: dilation and erosion.

Dilation entails scanning an image with a predetermined kernel, where the pixel at the anchor point is substituted with the maximum pixel value defined by the programmer. This process expands the white regions within a binary image.

In contrast, erosion functions as the inverse of dilation. It replaces the pixel at the anchor point with the minimum pixel value set by the programmer, effectively diminishing the thickness of white objects in a binary image. These operations enhance the quality of the binary image by removing noise and improving the distinction between characters and background, which is crucial for accurate character segmentation and recognition.

2.3. Edge Detection

For edge detection we used Sobel's Edge detection [15]. Sobel's edge detection is based on convolving the image I with two 3x3 kernels, Gx and Gy , to compute the gradient magnitude G at each pixel:

$$G=\sqrt{Gx^2+Gy^2} \quad (2)$$

Where Gx and Gy are the gradient approximations in the horizontal and vertical directions, respectively. These gradient approximations are obtained by convolving the image with the following kernels:

$$Gx=\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (3)$$

$$Gy=\begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

Sobel's method is effective in detecting edges while preserving important structural details. To optimize the Sobel edge detection, we apply non-maximum suppression to thin the edges and use a double threshold to identify strong, weak, and non-relevant pixels. After computing the gradient magnitude G, thresholding or other post-processing techniques can be applied to detect and highlight edges in the image.

2.4. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a machine learning method used for classification and regression tasks. It finds the nearest data points in the training set to a given input and predicts based on their majority class or average value. KNN is straightforward to implement and understand, making it popular for beginners and as a benchmark for other models. However, its performance can depend on factors like the choice of distance metric and the number of neighbors considered. For our research, we utilized the K-Nearest Neighbor (KNN) algorithm with k=3.

3. IMPLEMENTATION

To enable plate character recognition, our project is structured into four distinct phases: plate extraction, plate color determination, plate segmentation, and character recognition.

3.1. Number plate extraction

The initial step involves extracting the area of interest, which in our scenario pertains to the number plate. Various image processing techniques were employed to isolate the number plate region, as depicted in the flowchart presented in Figure 1.

To isolate the plate region from an image, we begin by converting the input image from RGB to grayscale, as depicted in Fig.2. The image is cropped and converted to a grayscale image as shown in Fig 3. This is done to simplify further processing steps and focus specifically on the relevant features of the vehicle, such as the number plate. Grayscale conversion reduces the complexity of the image by removing color information while retaining important structural details, making it easier to detect edges and perform subsequent image processing tasks.

Additionally, cropping helps to isolate the region of interest, minimizing unnecessary background information and improving the efficiency of the algorithm. Subsequently, the image undergoes blurring and various morphological operations such as erosion and dilation, illustrated in Fig. 4. The resulting grayscale image is then subtracted from the image obtained after erosion and dilation, yielding a processed binary image, as demonstrated in Fig. 5. Meanwhile, the pre-blurred image is utilized to detect edges, which are converted into another binary image. This second binary image is subtracted from the first, as depicted in Fig. 6. Subsequently, all contours are sent to an array and subjected to validation tests.

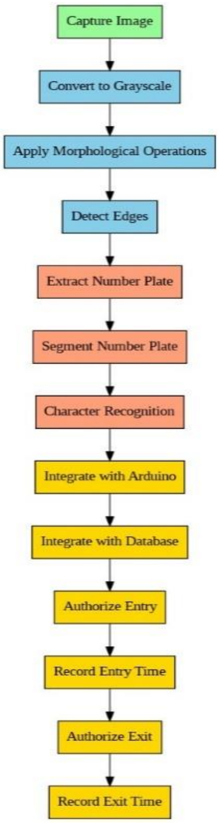


Figure 1. Flowchart of this research

Figure 2. Sample RGB Image of the car before preprocessing

Figure 3. Image of the car after cropping and grayscale conversion

The first test assesses contour area, eliminating those below a defined threshold to filter out small noises. The second test evaluates aspect ratio, considering the standard size of government-issued number plates but allowing for a range due to potential errors. Contours failing this test are discarded, except for the one with the largest area, which is likely to contain the vehicle's number plate after most potential candidates have been removed through cropping. The binary image of this selected contour, shown in Fig. 7, is then cropped for subsequent processing.

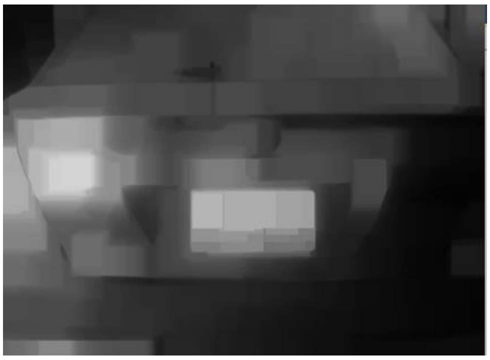


Figure 4. Image after morphological operations

Figure 5. Final Binary Image

Figure 6. Final Binary Image after subtraction

Figure 7. Binary image of the extracted number plate

3.2. Detemining the color of the plate

According to BRTA regulations, green-colored number plates are assigned to public commercial vehicles like CNGs (auto-rickshaws), buses, and taxis, whereas private vehicles have white plates. In our system, we extract the plate area from the original image and assess its RGB color spectrum. If the image exhibits a 'green' color, it indicates that the vehicle falls under the category of public commercial vehicles.

3.3. Number plate segmentation

The number plates in Bangladesh possess several distinctive features that were leveraged to enhance the system's efficiency and aid in the segmentation process:

- The upper part of the plate exclusively contains words (name of the region) and letters (vehicle class)
- The last character in the upper segment must always be one of the 41 characters as per BRTA regulations.
- Characters in the bottom half are consistently numeric.
- Fonts in the bottom segment typically appear larger and bolder compared to the characters in the upper segment.

After extracting the number plate, the image undergoes cropping to remove borders, reducing interference with the letters caused by border edges. Specifically, 5% is cropped from the top, and 1.25% from all other sides of the image. Typically, the extracted plate comprises two lines: the first line containing words or alphabets, and the second line containing only numbers. To adhere to this structure, the binary image undergoes preliminary adjustments for any skewing. Following this, the image is divided into two segments, as shown in Fig. 8.

Segmenting the image offers significant benefits by enabling the program to focus on individual segments during processing. This division of letters and numbers into separate sections allows the program to reduce the number of comparisons, thereby improving accuracy. It's important to highlight that the final character in the upper segment usually corresponds to a single Bengali alphabet, chosen from a set of 41 possible alphabets according to BRTA regulations.

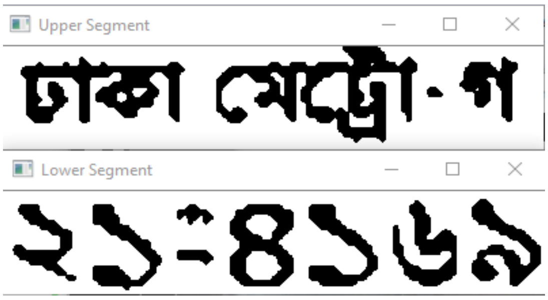


Figure 8. Image segmented into two parts.

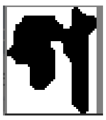


Figure 9. Last character from the top segment that determines vehicle class.

To enhance the algorithm's efficiency further, we isolate the last character (contour) from the first segment, specifically the contour at the right side. This separation ensures that these characters are solely compared with the 41 possible alphabets, as depicted in Fig. 9. Additionally, we determine if the number plate belongs to a public commercial vehicle class in a preceding step. This information enables us to compare the last character from the upper segment with a reduced number of samples, thereby enhancing the system's accuracy.

For the lower segment, our program initially checks if the lower segment contains six characters. If fewer than six characters are detected, it indicates that some characters have merged. Consequently, we measure the width of all contours, and if any contour significantly exceeds the median width, it is separated to ensure that the total number of contours in the lower segment remains six. The lower segment characters are usually Bengali numeric characters.

3.4. Character recognition

Contours identified within these segments are subjected to a validation process, where they must surpass a predefined minimum area threshold. This validation step is essential for filtering out any incidental marks that might appear on the plate. Contours that meet the validation criteria are then annotated with a bounding box, as illustrated in Figure 10.

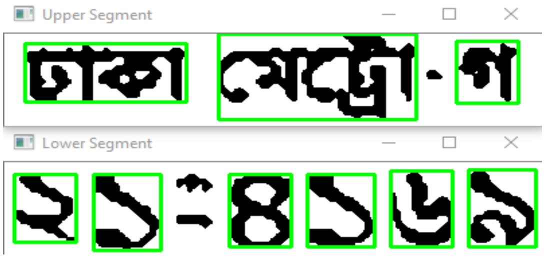


Figure 10. Effects of selecting different switching under dynamic condition

To identify each character properly, we employed the K-Nearest Neighbors (KNN) algorithm. Characters from the initial segments were matched with a list of terms, including city names, as well as keywords like 'মেট্রো', 'পুর', and 'খালী', often suffixes to certain city names. The last character from the upper segment was compared against the 41 possible alphabets specified by BRTA, denoting the vehicle's category. Subsequently, characters from the lower segment were compared against the numeric range of ten [0-9]. This segmentation reduced comparisons, enhancing accuracy.

The training process for the K-Nearest Neighbor (KNN) classifier involves preparing a comprehensive dataset of Bangla characters. The dataset includes a total of 5035 images of individual Bangla characters, numbers and complex keywords like 'মেট্রো', 'পুর', and 'খালী' which are often part of the number plates. For our research, we utilized the K-Nearest Neighbor (KNN) algorithm with k=3. For the feature extraction purpose, the pixel intensity values are used. Each character image is resized to a fixed size (32x32 pixels), resulting in a feature vectory of 1024 values.

3.5. Arduino and Database Integration

In addition to the image processing and character recognition steps outlined above, the system incorporates Arduino UNO and MySQL database integration for real-time application in automated parking systems. Upon successful recognition of the vehicle number plate, the system initiates the process of storing the recognized characters in the MySQL database. The integration between the image processing system developed in Visual Studio and the Arduino microcontroller allows seamless communication between the two components.

Once the characters are registered in the database, the Visual Studio application triggers a signal to the connected Arduino microcontroller. Upon receiving the signal, the Arduino activates a GREEN LED, indicating permission for the vehicle to enter the parking area. This functionality can be implemented at entry gates of parking facilities, ensuring that only authorized vehicles gain access.

Furthermore, the system records the entry time of the vehicle in the MySQL database. This data can be utilized to calculate the duration of the vehicle's stay in the parking lot, facilitating accurate billing for commercial parking spaces. Similarly, the system can be extended to manage exit gates, where the vehicle's exit time is recorded, allowing for efficient management of parking facilities. The integration of the whole system is illustrated in Figure 11.

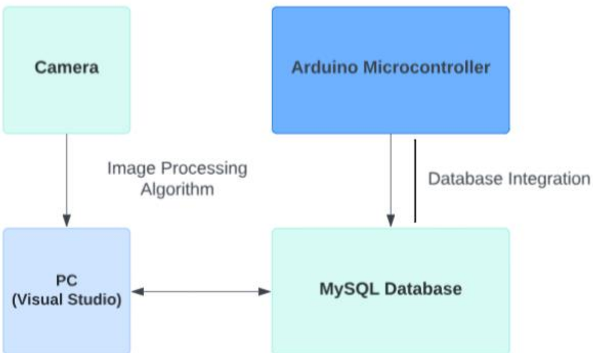


Figure 11. Integration of Vehicle Plate Recognition System with Arduino for Automated Parking

The figure illustrates the architecture of a vehicle number plate detection and recognition system with Arduino integration. The components include a camera for capturing images of vehicle number plates, a computer running Visual Studio with an image processing algorithm, an Arduino microcontroller, and a

MySQL database. The components are connected via USB/Serial connections, with the Arduino controlling a GREEN LED at the entry gate upon successful recognition of the vehicle number plate. Labels indicate the functions of each component and the connections between them.

Table 1 presents the database schema for the Parking_Record table, which is utilized to record vehicle entry and exit information in a parking facility. This schema includes fields for the recognized vehicle plate number (Plate_Number), the time of entry (Entry_Time), the time of exit (Exit_Time), and the corresponding parking fee (Parking_Fee).

Table 1. Database Schema

Field	Type	Description	
Plate_Number	VARCHAR(10)	Recognized vehicle plate number	"ঢাকা-মেট্রো-১২-৩৪৫৬"
Entry_Time	TIMESTAMP	Time when the vehicle entered the parking	2024-06-28 08:15:00
Exit_Time	TIMESTAMP	Time when the vehicle exited the parking	2024-06-28 08:16:00
Parking_Fee	DECIMAL(6,2)	Fee charged for parking duration	100

This integration enables efficient management of parking facilities, ensuring smooth entry and exit procedures while accurately recording vehicle movements for billing purposes.

4. RESULTS

Our system underwent real-life experimentation, with the majority of images captured using a cellphone on the streets of Dhaka City. These images were taken under diverse weather and lighting conditions, resulting in varying picture quality. A selection of test samples is shown in Fig. 12. Notably, 2% of the failed images produced unsatisfactory results primarily in identifying the last character in the top segment. Additionally, certain samples yielded poor results because the plates exhibited excessive noise and distortion. However, it is worth noting that the lower segments of all images performed impeccably across all test samples. The total number of test samples in this experiment comprised of 1007 images.

Certain Bangla characters were misrecognized due to their similarity in shape. Characters such as "ক" (ka) and "খ" (kha) or "গ" (ga) and "ঘ" (gha) can be challenging to distinguish, especially when the image quality is low or the characters are partially occluded. Images with poor resolution, significant noise, or blur tend to hinder the recognition process. The preprocessing steps, such as Gaussian blurring and thresholding, are less effective on such images, leading to errors in subsequent phases like edge detection and character segmentation.

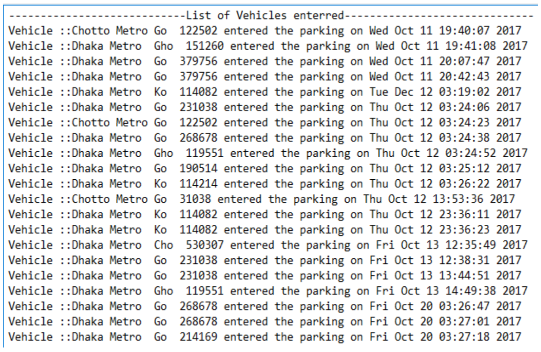
Table 2. Accuracy Percentage of our model and comparison with previous works

Algorithm	Accuracy %
Our Model	92.3%
[16]	88.8%
[17]	92.0%

The typical duration for both recognition and segmentation processes for the aforementioned table is recorded as 0.58 seconds on average, with a median time of 0.57 seconds. In comparison, the average time taken by the method detailed in a previous study [1] appears to be slightly longer, averaging at 0.64 seconds. This finding suggests that our method achieves slightly quicker processing times than the previously discussed approach. In scenarios with high vehicle traffic, such as busy urban intersections or large parking facilities, the processing time could become a bottleneck. If multiple vehicles need to be processed simultaneously, the cumulative delay might hinder real-time operation. While the current setup with Arduino UNO is sufficient for moderate traffic, upgrading to more powerful microcontrollers or dedicated processing units (e.g., GPUs) can significantly reduce processing time and enhance real-time performance.

Figure 11. Output of a sample image, (a) slightly blurred, (b) poor resolution and (c) slightly angled

A log file was generated from an experiment conducted with a selection of sample input images. This log file includes data on the entry time of vehicles, which can be utilized for setting up the database. The log entries depicted in Fig. 12 provides detailed information.



```
---------------------------List of Vehicles enterred-----------------------------
Vehicle ::Chotto Metro Go  122502 entered the parking on Wed Oct 11 19:40:07 2017
Vehicle ::Dhaka Metro  Gho  151260 entered the parking on Wed Oct 11 19:41:08 2017
Vehicle ::Dhaka Metro  Go  379756 entered the parking on Wed Oct 11 20:07:47 2017
Vehicle ::Dhaka Metro  Go  379756 entered the parking on Wed Oct 11 20:42:43 2017
Vehicle ::Dhaka Metro  Ko  114082 entered the parking on Tue Dec 12 03:19:02 2017
Vehicle ::Dhaka Metro  Go  231038 entered the parking on Thu Oct 12 03:24:06 2017
Vehicle ::Chotto Metro Go  122502 entered the parking on Thu Oct 12 03:24:23 2017
Vehicle ::Dhaka Metro  Go  268678 entered the parking on Thu Oct 12 03:24:38 2017
Vehicle ::Dhaka Metro  Gho  119551 entered the parking on Thu Oct 12 03:24:52 2017
Vehicle ::Dhaka Metro  Go  190514 entered the parking on Thu Oct 12 03:25:12 2017
Vehicle ::Dhaka Metro  Ko  114214 entered the parking on Thu Oct 12 03:26:22 2017
Vehicle ::Chotto Metro Go  31038 entered the parking on Thu Oct 12 13:53:36 2017
Vehicle ::Dhaka Metro  Ko  114082 entered the parking on Thu Oct 12 23:36:11 2017
Vehicle ::Dhaka Metro  Ko  114082 entered the parking on Thu Oct 12 23:36:23 2017
Vehicle ::Dhaka Metro  Cho  530307 entered the parking on Fri Oct 13 12:35:49 2017
Vehicle ::Dhaka Metro  Go  231038 entered the parking on Fri Oct 13 12:38:31 2017
Vehicle ::Dhaka Metro  Go  231038 entered the parking on Fri Oct 13 13:44:51 2017
Vehicle ::Dhaka Metro  Gho  119551 entered the parking on Fri Oct 13 14:49:38 2017
Vehicle ::Dhaka Metro  Go  268678 entered the parking on Fri Oct 20 03:26:47 2017
Vehicle ::Dhaka Metro  Go  268678 entered the parking on Fri Oct 20 03:27:01 2017
Vehicle ::Dhaka Metro  Go  214169 entered the parking on Fri Oct 20 03:27:18 2017
```

Figure 12. Log file showing the entries from the database

5. CONCLUSION

In conclusion, this paper introduces an innovative algorithm and techniques designed to facilitate the seamless detection and recognition of Bengali characters from vehicle plates. The Real-time testing in this research confirms the algorithm's robust performance, demonstrating its effectiveness in diverse weather conditions. By segmenting the plate into distinct regions, the algorithm achieves enhanced efficiency and accuracy, even under challenging image conditions such as slight angles, blurriness, and poor resolution. The successful integration of image processing algorithms with Arduino UNO further extends the applicability of the proposed solution, particularly in automated parking systems. The implementation of this algorithm addresses critical concerns in modern transportation management systems, offering a reliable and efficient solution for vehicle number plate detection and recognition. Additionally, our algorithm adheres to the standards set by the Bangladesh Road Transport Authority (BRTA), ensuring regulatory compliance and reliability in transportation management systems.

For long-term deployment, ongoing system maintenance is crucial to sustain performance over time. Regular updates and monitoring of the algorithm's performance will be essential to adapt to changing environmental conditions and maintain accuracy. Scalability considerations involve ensuring the algorithm can handle increasing volumes of data and processing demands as deployment scales across different locations and traffic conditions. Integration with existing infrastructure, such as database systems and real-time monitoring networks, will be pivotal for seamless implementation into broader transportation management frameworks.

To enhance future developments, incorporating advanced preprocessing methods such as adaptive thresholding and anisotropic diffusion could further improve image quality before further processing. Transitioning from traditional machine learning techniques like K-Nearest Neighbors (KNN) to deep learning models such as Convolutional Neural Networks (CNNs) could significantly boost recognition accuracy, especially in challenging scenarios like low-resolution images or heavily obscured characters.

REFERENCES

[1] R. A. Baten, Z. Omair, and U. Sikder, “Bangla license plate reader for metropolitan cities of Bangladesh using template matching,” in Proc.8th International Conference on Electrical and Computer Engineering, Dec 2014.

[2] I. E. Khatib, Y. Sweidan, S. -M. Omar and A. A. Ghouwayel, "An efficient algorithm for automatic recognition of the Lebanese car license plate," 2015 *Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE),* Beirut, Lebanon, 2015, pp. 185-189

[3] S. U. Khan, N. Alam, S. U. Jan, and I. S. Koo, "IoT-Enabled Vehicle Speed Monitoring System," *Electronics*, vol. 11, no. 4, p. 614, 2022, doi: 10.3390/electronics11040614.

[4] V. Pandey et al., "A Novel Method for Indian Vehicle Registration Number Plate Detection and Recognition Using CNN," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 1-23, Jan. 2024.

[5] A. Rakshe and N. Dongre, “An Efficient Methodology of Automatic Vehicle Number Plate Detection Using Deep Learning”, Int J Intell Syst Appl Eng, vol. 12, no. 12s, pp. 640 –, Jan. 2024.

[6] R. Antar, S. Alghamdi, J. Alotaibi, and M. Alghamdi, “Automatic Number Plate Recognition of Saudi License Car Plates*”, Eng. Technol. Appl. Sci*. Res., vol. 12, no. 2, pp. 8266–8272, Apr. 2022.

[7] R. S. Dangi, A. Kuvelkar, S. K. Maity, and S. Wandhekar, “Effic ient and robust indian number plate recognition through modified and tuned LPRNet,” in *ICDSMLA* 2020 , Lecture No., Singapore: Springer, 2022, pp. 115 – 129

[8] Kumar, J., Birudu, S., & Narkedamilly, L. (2021). "Automatic Vehicle Number Plate Recognition System Using Machine Learning." *IOP Conference Series: Materials Science and Engineering*, 1074, 012012. DOI: 10.1088/1757-899X/1074/1/012012.

[9] V. Uma Maheswari, R. Aluvalu and S. Mudrakola, "An Integrated Number Plate Recognition System through images using Threshold-based methods and KNN," *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Chiangrai, Thailand, 2022, pp. 493-497, doi: 10.1109/DASA54658.2022.9765218.

[10] G. N *et al*., "Automatic Number Plate Detection using Deep Learning," *2022 Smart Technologies, Communication and Robotics (STCR)*, Sathyamangalam, India, 2022, pp. 1-5, doi: 10.1109/STCR55312.2022.10009582.

[11] A. R. Youssef, F. R. Sayed and A. A. Ali, "A New Benchmark Dataset for Egyptian License Plate Detection and Recognition," *2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, Tianjin, China, 2022, pp. 106-111, doi: 10.1109/ACIRS55390.2022.9845514.

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

[13] Tom, R., Kumar, A., Shaik, S., Isaac, L.D., Tripathi, V., & Pareek, P. (2022). Car License Plate Detection and Recognition Using Modified U-Net Deep Learning Model. In 2022 8th *International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-6).

[14] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Syst. Man Cybern., vol. 9, pp. 62-66, 1979.

[15] S. Vijayarani and M. Vinupriya, "Performance analysis of Canny and Sobel edge detection algorithms in image mining," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 8, pp. 1760-1767, Oct. 2013.

[16] A.C. Roy, M. K. Hossen, D. Nag, “License Plate Detection and Character Recognition System for Commercial Vehicles based on Morphological Approach and Template Matching,” in Proc. *International Conference on Electrical Engineering and Information Communication Technology*, Sep 2016.

[17] A. Kashyap, B. Suresh, A. Patil, S. Sharma and A. Jaiswal, "Automatic Number Plate Recognition," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2018, pp. 838-843, doi: 10.1109/ICACCCN.2018.8748287.

BIOGRAPHIES OF AUTHORS

Niaz Ashraf Khan is a dedicated academic professional with over 5 years of experience in the field of Computer Science and Engineering. Currently serving as a Lecturer at the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh, Niaz holds a Master's degree in Computer Science and Engineering from North South University, Dhaka, Bangladesh, along with a Bachelor's degree from the same institution. his research interests lie in the intersection of Sound Signal Processing, NLP, machine learning, and deep learning. He can be contacted at email: niaz.ashraf@ulab.edu.bd

Md Ferdous Bin Hafiz is working at University of Liberal Arts Bangladesh as a Lecturer in the Department of Computer Science and Engineering. He completed his BSc. in Computer Science and Information Technology from Islamic University of Technology (IUT), Bangladesh. He obtained Master of Information Technology degree from Charles Sturt University, Australia. He has worked as a full-time lecturer for almost four years (two years at Northern University Bangladesh, one year and eight months in East Delta University, Chattogram). His research interests lie in the area of data science, machine learning and cyber security. He can be contacted at email: ferdous.hafiz@ulab.edu.bd.