An automated system to detect and recognize vehicle license plates of Bangladesh

Conference Paper · December 2017

DOI: 10.1109/ICCITECHN.2017.8281781

4 authors:

Sudipto Chandra
Shahjalal University of Science and Technology

Md. Al-amin Nowshad
Shahjalal University of Science and Technology

Mohammed J Islam
Shahjalal University of Science and Technology

Marium-E-Jannat Mukta
Shahjalal University of Science and Technology

An automated system to detect and recognize vehicle license plates of Bangladesh

Sudipto Chandra
Dept. of CSE, SUST
Sylhet-3114, Bangladesh
dipu.sudipta@gmail.com

Md. Al-amin Nowshad
Dept. of CSE, SUST
Sylhet-3114, Bangladesh
iamnowshad@gmail.com

M. Jahirul Islam
Dept. of CSE, SUST
Sylhet-3114, Bangladesh
jahir-cse@sust.edu

Marium-E-Jannat
Dept. of CSE, SUST
Sylhet-3114, Bangladesh
jannat-cse@sust.edu

***Abstract*—Bangladesh is a country in South Asia that uses Retro Reflective license plates. The plate has two lines with words, letters, and digits. An automated system to detect and recognize these plates is presented in this paper. The system is divided into four parts: plate detection, extraction, character segmentation and recognition. At first, the input image is enhanced using CLAHE and a matched filter specially designed for license plates with two lines is applied. Then tilt correction using Radon transformation, binarization and cleaning are performed. For character segmentation, mean intensity based horizontal and vertical projection is used. In recognition, we have used two different Convolutional Neural Network (CNN) to classify digits and letters. Tesseract OCR is used for district names. We have developed a dataset of over 400 images of different vehicles (e.g., private car, bus, truck etc.) taken at different times of day (including nights). The plates in this dataset are in different angles, including blurry, worn-out and muddy ones. On this dataset, the proposed system achieved a success rate of 96.8% in detection, 89.5% in extraction, 98.6% in segmentation and 98.0% in character recognition.**

***Index Terms*—Bangladeshi vehicles; Plate Detection; ALPR; ANPR; Character Recognition.**

I. Introduction

An automated system can do its job without any human intervention. In this paper, we present a system that takes an input image, detects positions of all license plates and recognize the plate number written on it.

The government of the country inaugurated retro-reflective license plate (Figure 1) as the standard in 2012, and made mandatory since 2016. Therefore, in our research, we only considered this type of plates.

Fig. 1. Standard Retro-Reflective License Plate

These standard license plates have two lines. The upper line contains a letter represents the vehicle type and area name of the issued plate. In case of metropolitan areas, the upper line has two words, but for general districts it only has the district name in one word. The bottom line contains 6 digits. These digits can be grouped together in packs of 2 and 4 digits. The first pack is usually the area code and the second one is vehicle identification number.

II. Literature Review

Research on automated license plate detection and recognition for vehicles in Bangladesh has been ongoing for less than a decade (judging by the paper published in various sources). Many of them used morphological operations for plate localization and reported accuracy around 70% to 95%. For character segmentation either horizontal and vertical segmentation, or the connected component analysis was used. And the recognition phase was generally done using Support Vector Machines or variations of Neural Networks.

[1] presented a simple system that takes images of car from a certain distance and angle, locates regions of interest by selecting a specific portion empirically, uses horizontal and vertical projection for segmentation and 4-layer neural network for recognition.

Both [2] and [3] proposed their method only up to plate detection. [3] used histogram checking and a vertical edge detection method to segment rows and columns as regions of interest, and achieved 93.2% accuracy, whereas [2] got 89.2% using a multi-stage approach to analyze vertical edge gradients from gray-scaled images.

A special feature of Bangla script called “Matra” was used in [4] to isolate the characters in plates. They claimed that their system works fine for images of various conditions. [5] also utilizes this feature. At first, they extracted the plate using morphological operations and fixed skew and rotation. After word segmentation, the “Matra” was detected and deleted from the segmented words to separate individual letters. Finally, a 4 layer NN was used as a classifier.

[6] used Hough line transformation to detect plates and a open-source OCR called SHABDAYON for recognition.

A customization of the system presented in [7] and [8] was used in [9] for plate detection. After segmentation, they extracted 25 features and classified them using a multi-layer perceptron.

[10] presented their ROID algorithm to detect regions, horizontal and vertical projection to segment characters, and an NN for recognition.

HSI color model for detecting ROIs was adopted in [11]. They also checked various geometric properties to pick out the most probable plate regions, and reported 85% accuracy in plate detection.

978-1-5386-1150-0/17/$31.00 2017 IEEE

CLAHE [12] was applied in [13] to process input images before extraction. Then 8-pixel connectivity, line-space and size properties were measured for segmentation. The recognition was done by calculating several distinctive and particular mathematical properties of segmented characters.

[14] followed conventional three stage system. For edge detection they used morphological operations- dilation and erosion, with structuring elements stretched along rows and columns subsequently. After extracting candidate regions, adaptive thresholding was applied to get binary image. Then connected component checking was done to segment characters from plate region. Finally, an SVM was used for character recognition.

[15] described the process of plate detection in various lighting conditions using noise removal, contrast enhancement, Local Counting Filter, connected component analysis, tilt correction etc.

[16] achieved 93% success in detection taking green color as priority and measuring geometrical properties of plates. This paper reported 98.1% success rate in segmentation 88.8% in Bangla text recognition.

Techniques such as Genetic algorithms [17], Fuzzy logic [18], Vector quantization [19] and Gabor filters [20] have been in use for plate detection, but was not ventured enough for Bangladeshi plates.

III. Our Framework

Figure 2 depicts the framework we proposed in this paper. It has four stages:

1) **Plate detection** discovers all regions of interest from given input image.
2) **Plate extraction** extracts the region of interest, corrects slanted or tilted plates and removes unnecessary noises around the text.
3) **Character segmentation** picks out district name, type letter, and digits from the clean plate image.
4) **Character recognition** calculates the probable plate numbers given the segmented characters.

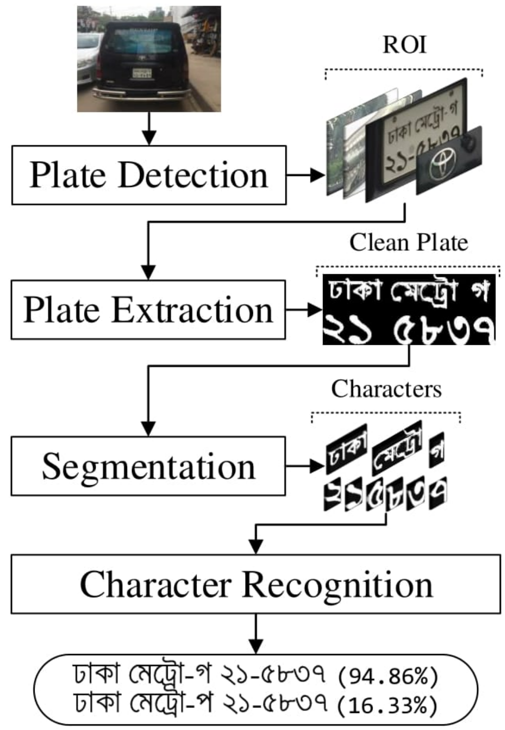

Fig. 2. The proposed framework

Fig. 3. Result of image enhancement (Frame = 60×30, Clip limit = 2.0)

A. Plate Detection

This stage identifies a set of region of interest (ROI) in the input image.

1) Input image: High-resolution camera is recommended for image acquisition. Recommended image size is 3840×2160.

2) Rescaling: The image is rescaled to fit within a frame of 640×480 keeping the aspect ratio intact.

3) RGB to Gray scale: After splitting the colored image into R, G, B components, Equation 1 is used to get the gray scale image that focuses on green colors.

$$Gray = 0.2125 \times R + 0.7154 \times G + 0.0721 \times B \quad (1)$$

4) Image enhancement: We used CLAHE (Clip Limited Adaptive Histogram Enhancement) [12]) with small-sized frame. Frame of size 60×30, and clip limit of 2.0 to enhance the image (Figure 3).

5) Edge Detection: The Sobel operator (Equation 2) is applied to the enhanced image followed by Otsu's thresholding to detect the edges of the image.

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

6) Matched filter: We use a modification to the mixture model presented by Joarder et al. [9] for ROI detection. The modified filter is disclosed in Equation 3. Fundamentally, it is a mixture of Gaussian kernels that gives strong response around plate like areas.

$$K_{x,y} = H_x \quad \text{for } 0 \le x < m, 0 \le y < n$$

$$H_x = \begin{cases} A.exp\left(\dfrac{-\left(x - \frac{m}{6}\right)^2}{0.2\sigma_x^2}\right), & 0 \le x < \dfrac{m}{3} \\ B.exp\left(\dfrac{-\left(x - \frac{5m}{12}\right)^2}{2\sigma_y^2}\right), & \dfrac{m}{3} \le x < 2\dfrac{m}{3} \\ A.exp\left(\dfrac{-\left(x - \frac{7m}{12}\right)^2}{0.2\sigma_x^2}\right), & 2\dfrac{m}{3} \le x < m \end{cases} \tag{3}$$

Fig. 4. A matched filter kernel (normalized image) of size $n \times m = 25 \times 12$ with constants $A = 0.003, B = -0.001, \sigma_x = 15, \sigma_y = 10$

After convolving the edge image with this kernel, we apply Gaussian blur with kernel of size 15×15 and apply Otsu's thresholding. A sample output is shown in Figure 5. The white regions in this image are possible locations of a plate.

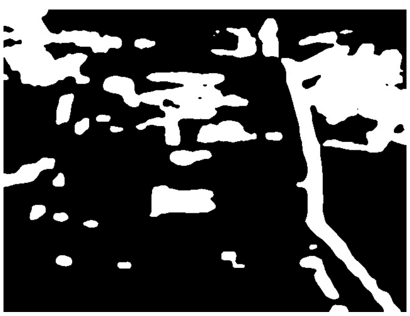

Fig. 5. Matched image after applying Gaussian Blur and Otsu's thresholding

7) Set of ROI: To extract and analyze each white portions, we apply contour analysis in the matched image. For each contour, the bounding box is calculated. We eliminate some of them that do not fit our constraints of height and width.

In our system, we keep a boxed region if $50 \le width \le 350$ and $20 \le height \le 150$, or discard it otherwise. The selected regions are then translated to the coordinates of the original input image from the scaled image. Figure 6 has some samples of extracted ROIs from Figure 5.

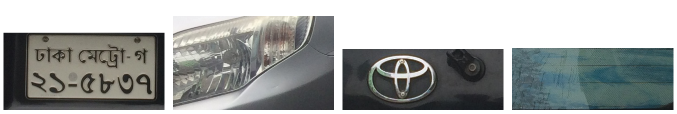

Fig. 6. Some regions of interest

B. Plate Extraction

For each ROIs obtained from the previous stage, we perform some special set of operations to further improve the plate quality.

1) Plate Image: We cut our region of interest from the input image and convert it to gray scale using Equation 1.

2) Contrast Enhancement: To enhance contrast, CLAHE with grid size of 8×8 and clip limit of 1.0 is applied. This produces some noise in the image. To reduce it, we use a bilateral filter of $diameter = 9$, $\sigma_{color} = 50$ and $\sigma_{space} = 75$.

3) Tilt correction: First, we use horizontal Sobel operator (Equation 4) to remove vertical edges. Canny edge detection algorithm is applied next to find out all vertical edges. Then, an operation known as Radon transformation [21] is used to find the dominant angle of the plate image.

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

If the rotation angle is more than 15°, we discard the image. Otherwise, we apply affine transformation to fix the rotation. Figure 7 shows some samples of this step.

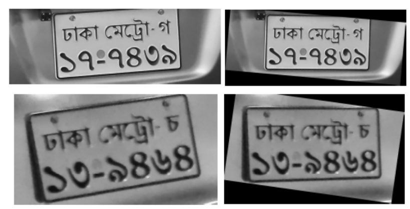

Fig. 7. Some samples of tilt correction

C. Cleaning Image

In this step, we convert the image into two colored binary image that has only white and black pixels. The binary image is passed through an algorithm which crops the borders surrounding the license plate. To remove all small dots and noises in the image, connected component analysis is used. Finally, the image is trimmed down to keep only plate numbers. Some examples are given in Figure 8.

Fig. 8. Original plate image and cleaned one are displayed side by side.

D. Character Segmentation

We followed a conventional method of character segmentation known as projection. At first, the horizontal projection is applied, then the vertical projection.

1) Horizontal projection: Using the mean intensity value of the pixels in the plate image across horizontal direction, we separate two lines.

2) Vertical projection: This process separates each line into individual characters. The character can either be a digit, letter, or word. To properly separate segments even if they are clumped together, we use different threshold of mean intensity across vertical direction and find out the best possible segments.

In this case, the upper line gives 2 or 3 segments containing 1 district name, and 1 type letter and the lower line gives 6 digits as its output. The segments are then numbered to keep track of their sequence. If the number of segments is not 8 or 9, we discard the region.

E. Character Recognition

We recognize letter, digit, and district name separately. The district names are not segmented and kept together as a whole unit to be recognized.

1) Preprocessing: The type letter and digits are trimmed and resized into 28×28 binary images. The district names are not rescaled.

Fig. 9. Some segments after preprocessing

2) Prediction: We pass the preprocessed segments to their respective recognizer and obtain prediction of classes with a confidence value. If there are 3 segments in the upper line, we recognize the second segment as ‘মেট্রো’(Metro).

We choose at most 3 classes of high confidence value for each segment (classes with $confidence < 10^{-3}$ are discarded). Next, we calculate all possible combinations of probable plate number by merging the segments along with their probability and keep maximum of 10 most probable predictions with probability of 10% or more. We show these plate numbers as output of our system.

F. Classifiers

Three different system is used for recognizing the (i) Digits, (ii) Letters, and (iii) Districts.

Among all possible vehicle types [22], we choose 22 letters which occur in the types of vehicles with proper license plates. These letters are: “অ ই উ ক খ গ ঘ চ ছ জ ঝ ট ঠ ড প ব ভ ম ল স শ হ”.

1) Digit and Letter Recognizer: For this purpose, we use two separate classifiers. They have almost the same configuration except for the number of neurons used in the output layer, as displayed in Table I.

2) District Recognizer: There are 64 districts in Bangladesh. License plates can be issued to any of these districts. Note that some district names are abbreviated, e.g.: “চট্ট” is used instead of “চট্টগ্রাম”.

To recognize the district names we use Tesseract OCR [23]. We take the output from the OCR and pass it into a string processing algorithm which gives the most probable district names with a confidence score. The algorithm is provided in Algorithm 1.

TABLE I
CONVOLUTIONAL NEURAL NETWORK FOR DIGITS & LETTERS

Layers	Parameters
Input layer	$unitsize = 28 \times 28 \times 1$
ReLU layer with 2D Convolution units	$kernel = 5 \times 5$, $outputs = 32$, $stride = 1$, $padding = 0$
Max Pooling layer	$kernel = 2 \times 2$, $stride = 1$, $padding = 0$.
ReLU layer with 2D Convolution units	$kernel = 5 \times 5$, $outputs = 64$, $stride = 1$, $padding = 0$
Max Pooling layer	$kernel = 2 \times 2$, $stride = 1$, $padding = 0$.
Fully connected Layer	1024 neurons
Output layer	10 neurons for digits, 22 for letters.

Algorithm 1 Finding district with a confidence score.
Input: Text from OCR
Output: District names with confidence score (at most 3)
Initialization :
$Districts =$ Array containing list of district names
LOOP Process
for $i = 1$ to $len(Districts)$ **do**
    $D =$ Edit-distance between the input and $Districts[i]$.
    $confidence = 1 - D/10$
end for
$R =$ Pick at most 3 items and at least one item with highest confidence score.
return R

IV. PREPARING THE CLASSIFIER

A. Dataset

The initial dataset is generated algorithmically from texts using various fonts and morphological operations such as: Dilation, Erosion, Top Hat, Affine etc. Additionally, some segmented characters from our system was selected, labeled manually, and included in the dataset.

Each object is rescaled to 28×28 and stored as Bitmaps. Finally, the images are converted to compressed binary data for training and testing (15% for testing). The labels are stored into separate compressed binary files. Table II shows the size of the final dataset. Figure 10 shows some sample objects. Note that we use the Tesseract OCR for district names, which does not need any additional training.

TABLE II
DATASET FOR TRAINING THE CLASSIFIER

Dataset	Total Objects	For training	For testing	Classes
Digits	11,190	9,511	1,679	10
Letters	49,170	41,794	7,376	22

B. Training

We trained our networks using Adam optimizer with dropout probability of 75%. In each step, the network is trained on only one item from the dataset. Figure 11 and 12 show the progress. After training, the final model was stored to be used later for prediction.

Fig. 10. Some sample objects from the dataset

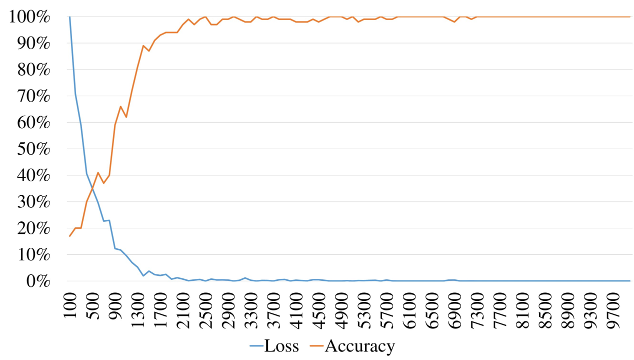

Fig. 11. Training of CNN for digit classification. After 10,000 steps it achieved 100% accuracy on the test dataset with 0% probability error.

V. Experiments

A. Testing data

To test our system against real life scenarios, we have chosen the test dataset carefully. More than 600 images of different vehicles were captured. They are private cars, minibus, bus, truck, lori etc. Both day and night time images were acquired using different types of camera. Finally, we picked out 405 images that correctly represent all kinds of scenarios like- slanted plates, blurry image, muddy or worn-out plate numbers, uneven illuminations etc.

B. Results of testing

Despite the conditions of our dataset, we have achieved a good success rate in plate detection (Table III). Figure 13 shows some plates that were detected correctly which we thought would fail.

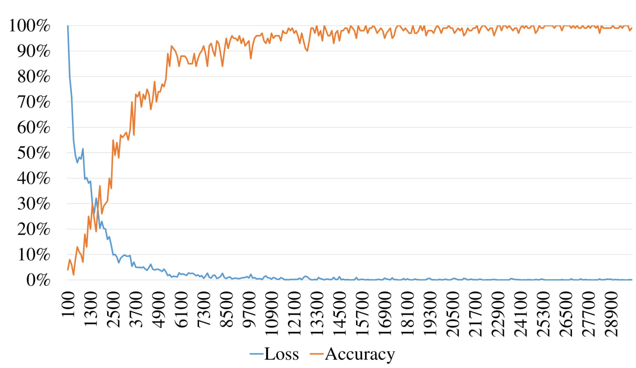

Fig. 12. Training of CNN for letter classification. After 100,000 steps it achieved 99.93% accuracy on the test dataset with 0% probability error.

Fig. 13. License plates in critical conditions detected by the system

Figure 14 displays the final result after character recognition for two plates.

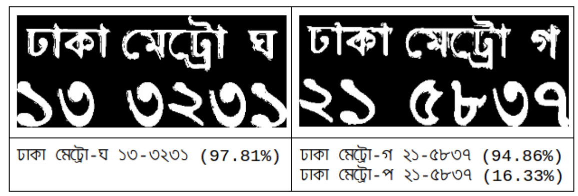

Fig. 14. Some recognized plate with probability

C. Accuracy

The success rate of our system that is given in Table III.

TABLE III
ACCURACY OF OUR SYSTEM IN VARIOUS STAGES

Stages	**Accuracy**	**Number of samples**	
		Passed	Total
1. Detection	96.8%	392	405
2. Extraction	89.5%	351	392
3. Segmentation	98.6%	346	351
4. Recognition	98.0%	339	346
(a) Digit	99.9%	2074	2076
(b) Letter	98.5%	341	346
(c) District	99.7%	345	346

D. Cases of failures

The detection fails for the images with heavily slanted plates, blurry and low-resolutions images, too noisy background, non-standard or worn-out license plates etc. Some samples of failure are given in Figure 15.

Fig. 15. Plate detection did not work for these images

For some plates like in Figure 13, extraction fails to correct rotation. There are also some cases where some small part of character body (e.g., dots) are removed during cleaning. If two characters are clumped together, or there exists some uncleaned white dots, it sometimes fails to be segmented properly. Most cases of failure in character recognition occurs when there is too much noise surrounding the characters or some parts of them are missing as shown in Figure 16.

Fig. 16. Some characters that fail recognition

E. Comparison

As the dataset used for experimentation is not the same for other described system, the actual comparisons with them can not be done veraciously. Therefore, our approach appears to be outperformed in some areas compared to others. However, we present the accuracy claimed by various papers in Table IV.

TABLE IV
CLAIMED ACCURACY OF OTHER SYSTEMS

	Success Rate(%)			
	D	**E**	**S**	**R**
Proposed	96.8	89.5	98.6	98.0
[5]		84	-	80
[6]	88		77	62
[9]		92.1	97.53	84.16
[14]		93.1	98.1	99.2
[24]	97.6		90.7	97.9

D: Detection, **E**: Extraction, **S**: Segmentation, **R**: Recognition
Note: Dataset and testing process was not same for these papers

After lots of experimentation, we observed that the sequence of operations we devised in our system performs best for our dataset. As the dataset were chosen to represent most of the cases of real life scenario, we can safely claim that our system will perform much better in a controlled environment, where camera is fixed and the images are fed from a live feed instead of still pictures. The method we used in recognition system is a novel approach which recognizes each segments separately.

VI. CONCLUSION

Recognition of license plate characters depends on segmentation, which in turn depends on the good quality clean images. To get clean and good quality images, the plate detection and extraction should be more accurate. We have achieved success for most of the easily readable and clean plate images but the highly slanted plates. The future research in this area should be focused on how to improve plate extraction for skewed or tilted plates in various illuminations and weather conditions.

REFERENCES

[1] M. S. Mashuk, M. A. Majid, N. Basher, and T. R. Rahman, "Automatic detection of bangla characters in bangladeshi car registration plates," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on*. IEEE, 2010, pp. 166–171.

[2] S. Saha, S. Basu, M. Nasipuri, and D. K. Basu, "License plate localization from vehicle images: An edge based multi-stage approach," *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp. 284–288, 2009.

[3] A. A. Moustafa and M.-I. R. M. Jaradat, "A new approach for license plate detection and localization: Between reality and applicability," *International Business Research*, vol. 8, no. 11, p. 13, 2015.

[4] R. A. Baten, Z. Omair, and U. Sikder, "Bangla license plate reader for metropolitan cities of bangladesh using template matching," in *Electrical and Computer Engineering (ICECE), 2014 International Conference on*. IEEE, 2014, pp. 776–779.

[5] A. K. Ghosh, S. K. Sharma, M. N. Islam, S. Biswas, and S. Akter, "Automatic license plate recognition (alpr) for bangladeshi vehicles," *Global Journal of Computer Science and Technology*, 2011.

[6] M. R. Amin, N. Mohammad, and M. A. N. Bikas, "An automatic number plate recognition of bangladeshi vehicles," *International Journal of Computer Applications*, vol. 93, no. 15, 2014.

[7] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," *Image and Vision Computing*, vol. 27, no. 8, pp. 1134–1142, 2009.

[8] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377–392, 2006.

[9] M. M. A. Joarder, K. Mahmud, T. Ahmed, M. Kawser, and B. Ahamed, "Bangla automatic number plate recognition system using artificial neural network," *Asian Transactions on Science & Technology (ATST)*, vol. 2, no. 1, pp. 1–10, 2012.

[10] M. M. Hasan, "Real time detection and recognition of vehicle license plate in bangla," 2011.

[11] K. Deb, M. K. Hossen, M. I. Khan, and M. R. Alam, "Bangladeshi vehicle license plate detection method based on hsi color model and geometrical properties," in *Strategic Technology (IFOST), 2012 7th International Forum on*. IEEE, 2012, pp. 1–5.

[12] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[13] N. A. Siddique, A. Iqbal, F. Mahmud, and M. S. Rahman, "Development of an automatic vehicle license plate detection and recognition system for bangladesh," in *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*. IEEE, 2012, pp. 688–693.

[14] M. A. Uddin, J. B. Joolee, and S. A. Chowdhury, "Bangladeshi vehicle digital license plate recognition for metropolitan cities using support vector machine," in *Proc. International Conference on Advanced Information and Communication Technology*, 2016.

[15] S. Azam and M. M. Islam, "Automatic license plate detection in hazardous condition," *Journal of Visual Communication and Image Representation*, vol. 36, pp. 172–186, 2016.

[16] A. C. Roy, M. K. Hossen, and D. Nag, "License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching," in *Electrical Engineering and Information Communication Technology (ICEEICT), 2016 3rd International Conference on*. IEEE, 2016, pp. 1–6.

[17] S. Yoshimori, Y. Mitsukura, M. Fukumi, and N. Akamatsu, "License plate detection using hereditary threshold determine method," in *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 2003, pp. 585–593.

[18] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 1, pp. 42–53, 2004.

[19] R. Zunino and S. Rovetta, "Vector quantization for license-plate location and image coding," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 159–167, 2000.

[20] F. Kahraman, B. Kurt, and M. Gökmen, "License plate character segmentation based on the gabor transform and vector quantization," in *ISCIS*. Springer, 2003, pp. 381–388.

[21] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 1004–1008, 2005.

[22] http://www.brta.gov.bd/newsite/en/vehicles-class/.

[23] R. Smith, "An overview of the tesseract ocr engine," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 629–633.

[24] A. Rabee and I. Barhumi, "License plate detection and recognition in complex scenes using mathematical morphology and support vector machines," in *Systems, Signals and Image Processing (IWSSIP), 2014 International Conference on*. IEEE, 2014, pp. 59–62.