Deep Learning-Based Bangladeshi License Plate Recognition System

Conference Paper · November 2020

DOI: 10.1109/ISMSIT50672.2020.9254748

6 authors, including:

Tanmoy Sarkar Pias
The University of Asia Pacific
20 PUBLICATIONS **70** CITATIONS
SEE PROFILE

Tanjina Helaly
University of Asia Pacific
7 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Nymur Rahman
East West University (Bangladesh)
1 PUBLICATION **7** CITATIONS
SEE PROFILE

Deep Learning-Based Bangladeshi License Plate Recognition System

Md. Mesbah Sarif
Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
16101128@uap-bd.edu

Tanmoy Sarkar Pias*
Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
tsr@uap-bd.edu

Tanjina Helaly
Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
tanjina@uap-bd.edu

Md. Sohel Rana Tutul
Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
16101061@uap-bd.edu

Md. Nymur Rahman
Computer Science and Engineering
University of Asia Pacific
Dhaka, Bangladesh
16101078@uap-bd.edu

***Abstract*—A License Plate Recognition (LPR) is a technique to automatically identify and recognize the license plate of a vehicle. In today's world where we want to do everything in no time, automated LPR has opened a new door in Intelligent Traffic System. With the help of ALPR, we can implement many smart systems starting from a parking facility to crime investigation. For effective License Plate Recognition, accurate License Plate Detection and License Plate Segmentation are crucial. Many existing algorithms are using different image processing and machine learning techniques for detection, segmentation, and recognition. These algorithms are highly computational extensive and somewhat unreliable. In our paper, we have proposed an end-to-end license plate recognition system for Bangladeshi vehicles. We have divided our system into three different phases. The first phase is license plate detection and localization. For that, we have used YOLOv3. In the second phase, we have implemented a custom segmentation algorithm specifically for Bangladeshi license plates which is robust and computationally fast. In the final phase, we have implemented a character recognition model with convolutional neural network (CNN). This recognition system has achieved 97.5% accuracy. Moreover, we have built a diversified dataset with 2000 images where we have tried to capture the environmental factors**.

Keywords— image processing, deep learning, CNN, LPR, YOLO

I. INTRODUCTION

Automated License Plate Recognition is the first step for implementing many smart applications in our Traffic system. A license plate is a unique identifier of a vehicle. A registration number can provide the owner's information as well. Hence, LPR can be used to solve different criminal activities related to vehicles such as car theft, kidnapping, drunk & drive, reckless driving, hit & run, illegal vehicles and many more.

Typically, an ALPR system consists of 3 phases: i) License Plate Detection, ii) Character Segmentation and iii) Character Recognition. The correctness of any phase depends on the correctness of previous phases. Many existing algorithms are using different image processing and machine learning techniques e.g. deep learning, neural network, CNN, Region-based Convolutional Neural Networks(R-CNN), OpenCV, template matching, connected components at each stage of ALPR and most of these algorithms are computationally heavy and also need an expensive computational system to get the result faster.

We have proposed a robust license plate recognition system with image processing and deep learning. Our proposed end-to-end Bangladeshi license plate recognition system has three phases such as license plate detection and localization, license plate segmentation, and segment recognition.

Contributions of this paper are:

a. We have provided a state-of-the-art License Plate Recognition system for Bangladeshi vehicles.
b. We have created a custom segmentation algorithm for the Bangladeshi license plate which can segment any visually separable characters from the license plate.
c. We have implemented a CNN model for the segment recognition which achieved 97.5% accuracy.
d. We have introduced a rich dataset of the Bangladeshi license plate that contains 2000 images that are captured in different environmental conditions.
e. We have preprocessed the raw dataset into the YOLOv3 format for the detection task.
f. We have introduced a dataset with the segments from the license plate for the recognition task.

The rest of the paper is organized as follows: Literature review is discussed in section **II**. An overview of the Bangladeshi license plate is described in section **III.** Overview, description, working flow, and breakdown of our system are given in this section I**V.** The result is provided in this part of the paper **V**. Conclusion and discussion on our paper are given in this section **VI**. The rest of the part is the reference.

II. Literature Review

M M Shaifur Rahman et al. proposed a license plate character recognition model In 2017, which achieved very good accuracy [1]. However, they only implemented the character recognition part with a CNN model, and their model accuracy 88.67%. They used around 1700 images for training. As they focused only on the recognition task, they had to manually detect the image of the license plate from the vehicle and crop it accordingly. Not only that, but they also had to manually crop the characters to build their training dataset. So, they had to undergo huge labor to build the dataset. They worked with 16 classes but they didn't mention the name of classes. However, they broke down different segments like "Dhaka" and "Metro" which consists of multiple characters. Because an end-to-end license plate recognition model can be divided into three modules. The first module will detect the license plate from an image or a video. Upon detection, the license plate will be segmented (cropped) from the image. The second module will extract the characters or segments from the license plate image. And finally, all the extracted characters will be recognized in the last module. In this process, the segments "Dhaka" and "Metro" will be extracted as a whole segment. As a result, their recognition model fails.

Sohaib Abdullah et al. proposed a complete license plate recognition system for Bangladesh Vehicles [2]. In the first stage, they used YOLOv3 to detect the license plate from an image of a vehicle. This detected license plate is fed into the second stage for character detection and localization. However, this system is only for the Dhaka Metropolitan area as they didn't use other metropolitan license plates. So, they didn't detect the "Dhaka" in this stage. They focused on the 6 digits in the second line of the license plate and the character at the end of the first line which represents the vehicle type. In the final stage, they implemented ResNet-20 for the digit and character recognition. They achieved 92.7% accuracy in the digit and character recognition stage.

Raiyan et al. proposed a license plate recognition method with template matching in 2014 [3]. However, they only focused on the character and digit recognition task. They used a special feature of Bangla character called "matra". They mainly used image processing techniques here. They showed some test cases but they didn't make any general conclusion which makes it image specific. However, they didn't mention any dataset. They only used a small number of images to perform the experiment which lacks generalization. Moreover, they didn't mention any detection system for the license plate.

Prashengit et al. proposed a Bangladeshi license plate recognition system using image processing and machine learning [4]. They introduced a license plate dataset with 2800 images. They used an image processing technique called shape validation to detect the license plate from an image. Then they again used image processing for tilt correction. After that, they used connected components to segment the characters in the license plate. They extracted the connected components which represent different characters of the license plate. They implemented the recognition task with the Adaboost classifier. For recognition, they used two main features which are Histogram of Gradient (HOG) and Local Binary Pattern (LBP). Their proposed algorithm achieved 97.3% accuracy which seems to be the current state-of-the-art model. However, this system mostly depends on several image processing techniques like binarization, morphological analysis, shape detection, shape validation, ROI extraction, tilt correction, HOG, and LBP. As a result, this system is somewhat time-consuming.

Nazmus Saif et al. proposed an end-to-end system for Bangladeshi license plate recognition [5]. However, their paper has not releveled specific information on their dataset. So, it is very hard to evaluate or compare their work. Moreover, they claimed to achieve 99.5% accuracy on their dataset. They used the YOLOv3 for both license plate detection and recognition which is a very naïve selection for this type of task. But before concluding their work, we recreated their model to evaluate their claim. Unfortunately, we could not achieve that accuracy using YOLOv3 on our dataset. We think they also encounter this problem and added this issue to the limitation. In conclusion, their model achieved a very high accuracy because their dataset was not diverse enough and cannot be implemented for real-life scenarios. Recently, a sensor-based vehicle recognition system has also been proposed [9].

III. Bangladeshi Standard License Plate

To drive a vehicle on the Bangladeshi road, the vehicle must have a license plate verified by BRTA (Bangladesh Road Transport Authority). BRTA is the only authority to give license plates or to give road permits. BRTA started approving license plates since 1973.Previously, it looked different. In 2012 BRTA started giving a new type of license plate called a digital license plate. The public transport vehicles have black text on a green background and private transport vehicles have black text on white background in the digital license plate.

We can divide the license plate into two rows. In the first row, the first segment represents the city name, the second one is always metropolitan and the third character represents the vehicle category. In the second row, the characters represent digits in the Bangla language. Six digits are divided into two segments. The first segment consists of two digits which mean the class number of the vehicle and the second segment has four digits separated by a dash (-) from the class number. These four digits are the vehicle number. We will try to segment city number, metropolitan, vehicle number, and all six digits.

Figure 1. Private (left) and Public (right) License Plate

The general format of license plates is "city code - vehicle class letter - class number - vehicle number". For example: "DHAKA METRO-D-11-9999". The "DHAKA METRO" field represents the city name in Bengali letters, the "D" field represents the vehicle class in Bengali letters, the "11" field represents the class number in Bengali numerals and the "9999"

field represents the vehicle number of the vehicle in Bengali numerals. Here the characters are printed into 2 separate lines. The first line contains “DHAKA-D” and in the second line it contains “11-9999”. The city name can vary depending on its registration place. So, the city name will have multiple classes. The vehicle class which is represented by one character is taken from a set of {অ, ই, উ, এ, ক, খ, গ, ঘ, ঙ, চ, ছ, জ, ঝ, ত, থ, ঢ, ড, ট, ঠ, দ, ধ, ন, প, ফ, ব, ভ, ম, য, র, ল, শ, স, হ}. The digits can be 0-9 but in Bangla language which is {০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯}.

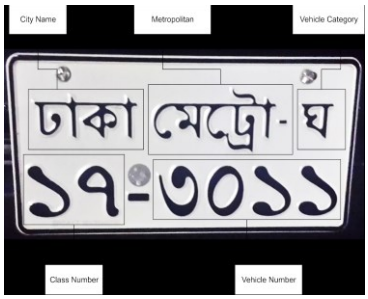


Figure 2. Bangladeshi Standard License Plate

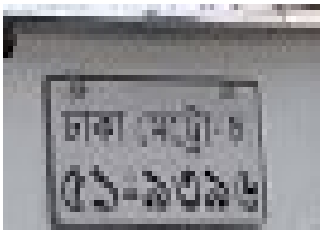
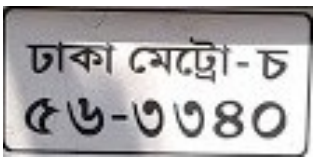
IV. Methodology and Implementation

A. Detection and Localization

In our proposed system we have a subsystem to detect and locate the license plate from the image. For this, we have used You Only Look Once (YOLOv3) which is the state-of-the-art object detection model. YOLO is a clever neural network for object detection in real-time. YOLO takes a completely different approach to detect an object by looking at the object only once. So, it is a very fast and effective real-time detection model. YOLO divides the whole image into N*N cells. Each cell is responsible for making 5 bounding boxes. YOLO generates a confidence score according to the presence of any object in the bounding box. Bounding boxes with low confidence scores are rejected. In this architecture, we have trained the YOLOv3 model with 2000 bounding box images of the license plate. We have used the LabelImg tool for making bounding boxes. The LabelImg tool outputs an XML file for every image and this XML file contains the coordinate of the license plate. Then the output file is converted to YOLOv3 format. After the data preparation, the training process is started with 2000 images which have different angles and light conditions. This custom YOLOv3 model predicts only one class in an image using a single convolutional neural network.

As a result, the process is quite fast. Surprisingly, this model has a pretty good prediction on the various tilted license plates. It can also detect tilted license plates from the low-resolution pictures. If the license plate is located in the far distance, the cropped image becomes a very low-resolution image. So the picture quality should be higher to get the long distant license plate. It works fine in shaded images too.

TABLE I. Detection Result

Detection	Segmentation
	
	
	
	
	 

To train this model we have used 2000 training images and 400 validation images. Other parameters of the training process is given in the table II. We have worked on an environment with CPU of Intel i7 7th gen processor, RAM of 8 Giga Bytes DDR4 2400 MHz, and a GPU of NVIDIA GTX GeForce 1070. The GPU accelerated the training process to a great extent.

TABLE II. YOLOv3 MODEL CONFIGURATION

Model Attributes	Values
Train Dataset	2000
Validation Dataset	400
Initial Epoch	50
Initial Batch Size	8
Secondary Epoch	100
Secondary Batch Size	16
Optimizer	Adam
Learning Rate	1e-3
Validation split	0.2

A. Character Segmentation

The algorithm proposed in [11] starts by taking a color image. Then, it converts the image into an 8-bit binary image. After that, the binary image is mapped into 0 and 1 where 0 represents the white region or the background, and 1 represents the black regions or the characters. Then, our greedy algorithm starts working on it to segment the characters. Our algorithm maintains a modified BFS traversal and flood fill method. When it reaches an unvisited black pixel, it stores its coordinates in a queue. Then the coordinates are popped from the queue and the algorithm checks it's 8 neighbor pixels and tries to find an unvisited black pixel. If it finds an unvisited black pixel then it stores its coordinates in a queue and the value of the row and column are stored in the R and C list respectively. This process repeats until the queue is empty. After that, the minimum and maximum values are taken from R, C, and then the R, C is cleared. Thus, we get the first segment. And this whole process is repeated as the algorithm is traversing the whole picture and thus, we get all the segments of the LP.

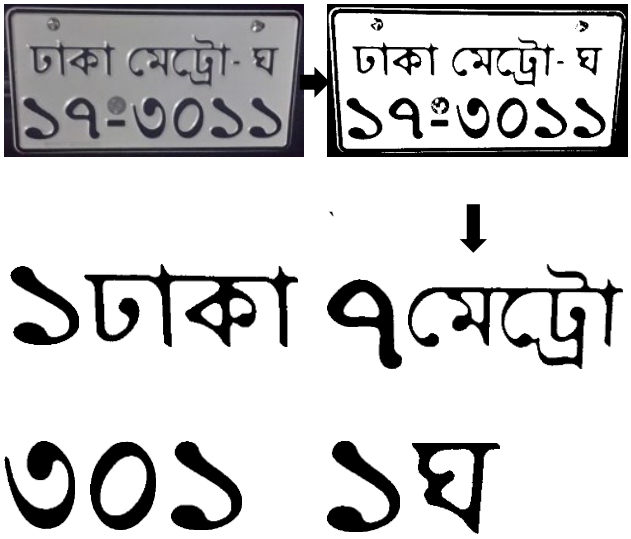


Figure 3. License plate segmentation

Algorithm 1: Greedy Graph-based Segmentation [11]

1. **Input** 24-bit RBG color image of the license plate
2. **Output:** Set of characters from the license plate
3. Convert the image into an 8-bit grayscale image
4. Convert the grayscale image to a binary image composed of 0 and 255
5. Map 0 with 1 and 255 to 0. 1 means a black pixel and 0 means a white pixel
6. For **c** in column
7. For **r** in row
8. If image[r][c] == 1 and visited[r][c] == 0:
9. Visited[r][c] = 1
10. Push (r, c) into queue Vertex, r into list R, and c into list C
11. While queue Vertex!= empty:
12. Check for unvisited black from adjacent 8 neighbor of the pixel (r, c)
13. If found push the coordinate (r,c) into the queue Vertex, r into R, and c into C
14. max_c = maximum of C
15. max_r = maximum of R
16. min_c = minimum of C
17. min_r = minimum of R
18. Clear C, Clear R
19. Push (min_c, min_r, max_c, max_r) into a list character_box
20. For each element in character_box:
21. Crop the rectangular segment using (min_c, min_r) and (max_c, max_r) which represents the top left corner and bottom right corner of a rectangle.

B. Character Recognition

The final module of our end-to-end system is the recognition of the character or segments. As we have seen in the previous section, we get different segments from the license plate image. For the recognition task, we have implemented a convolutional neural network (CNN) model. CNN has also been used in complex medical image recognition tasks and performs well enough [10].

Finally, in this dataset, we have tried to cover all the different classes. But for recognition with CNN, we need a good number of images of each class. We have maintained two different versions of our raw dataset. One dataset is for license plate detection and localization which contains around 2000 images collected from the streets in different environmental conditions. However, it was not required to use the whole dataset for segment recognition task. Moreover, one image can have multiple vehicles with visible license plates. There are also a number of images where the license plates are not entirely visible but detected by YOLOv3. So, make the recognition system more robust, a diverse subset of the total dataset is used. In brief, this recognition dataset contains the segments which are the output of the segmentation module. An overview of our recognition dataset is given in the table III.

TABLE III. DATASET FOR RECOGNITION

Class Name (Original)	Class Name (English)	Train Dataset	Validation Dataset
ঢাকা	Dhaka	197	50
.মেট্রো	Metro	194	40
০	0	101	30
১	1	215	50
২	2	169	33
৩	3	161	22
৪	4	104	12
৫	5	137	22
৬	6	85	12
৭	7	117	26
৮	8	100	23
৯	9	81	17
ক	Ka	2	1
খ	Kha	8	1
গ	Ga	115	22
ঘ	Gha	100	3
চ	Cho	8	1
থ	Tha	23	2
		Total: 1917	Total: 367

TABLE IV. RECOGNITION OUTPUT

Input Segments	**Recognition Output**
১	1
ঢাকা	Dhaka
৭	7
মেট্রো	Metro
৩	3
০	0
১	1
১	1
ঘ	Gha

Here as we couldn't cover all types, we took the 13 classes which are "Dhaka", "Metro", 0-9, "Ga (গ)" and "Gha (ঘ)". The segmentation process outputs every single number of digits as well as characters. We run a four-layer multiclass CNN model to recognize each digit and character segments. This model has achieved a very good accuracy of 97.50% on the proposed dataset.

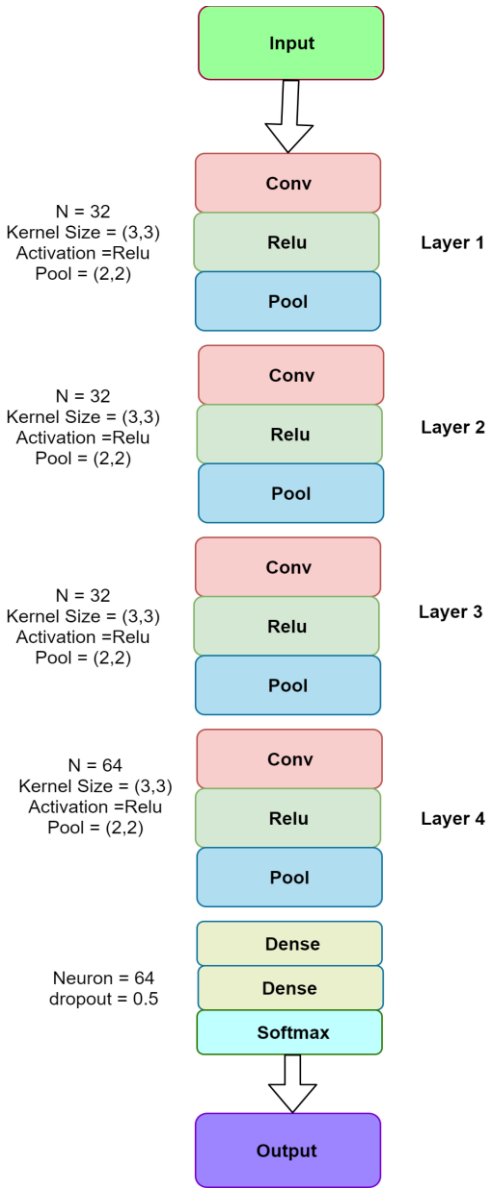


Figure 4. CNN model for the character recognition

V. RESULT AND ANALYSIS

We have used a diversified dataset for training and testing our model. We have achieved an accuracy of 97.5% in the character recognition part which is still now the highest accuracy in this field. We have compared our system with previously existing systems and our system has proved to be better.

TABLE V. ACCURACY COMPARISON

Paper Name	Sample Images	Recognition Accuracy
Automatic License Plate Recognition (ALPR) for Bangladeshi Vehicles [7]	Testing: 300	80%
Bangla Automatic Number Plate Recognition System using Artificial Neural Network [8]	Testing: 248	84.16%
Line Segmentation and Orientation Algorithm for Automatic Bengali License Plate Localization and Recognition [6]	Testing: 119	84.87%
Automatic License Plate Recognition System for Bangla License Plates using Convolutional Neural Network.[1]	Training:1700	88.67%
YOLO-Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City.[2]	Total: 1, 500	92.7%
Bangladeshi License Plate Recognition Using Adaboost Classifier[4]	Total: 2800	97.3%
Our Proposed Method	**Training: 1917 Testing: 367**	**97.5%**

Here our main competitor is the model of paper [4]. It achieved 97.3% accuracy with a large dataset. But we have achieved better accuracy with a relatively small and diverse dataset. So, it is valid to claim that our proposed model has the best accuracy among all the proposed systems.

VI. CONCLUSION

We have implemented a complete license plate recognition system with a greedy graph-based segmentation approach. The main novelty of our system is that it has a new custom greedy graph-based segmentation module. This algorithm can be applied to other segmentation problems with a little modification. But the limitation of this algorithm is that this can segment non-overlapping characters and cannot segment overlapped segments. Still, it is the state-of-the-art approach for segmentation of Bangladeshi License plate as there is literary no other algorithm with this level of performance. This algorithm can segment any visually recognizable characters from a license plate. The main motivation of this paper is to implement a robust segmentation algorithm specific for the Bangladeshi license plate recognition task. And we are successful in doing so. Our proposed algorithm is far accurate from traditional and state-of-the-art OpenCV segmentation algorithms. We don't claim this algorithm to be a generalized segmentation algorithm and our purpose is not doing so. The most important factor of this algorithm is that we don't need a detection phase to segment the registration numbers from the license plate. Moreover, we created our dataset in a way that it can capture the real-life scenarios. We have captured these images from a position where a real CCTV camera can be placed. For this reason, most of the images are both vertically and horizontally tilted. At the time of designing our segmentation algorithm, we have emphasized vertical and horizontal cases, we are glad that our novel segmentation algorithm can segment from any tilted images. Finally, this end-to-end Bangladeshi license plate recognition system has the highest overall accuracy.

ACKNOWLEDGMENT

We would like to thank the Institute of Energy, Environment, Research, and Development (IEERD, UAP) and the University of Asia Pacific for financial support.

REFERENCES

[1] Rahman, M. M., Mst Shamima Nasrin, Moin Mostakim, and Md Zahangir Alom. "Bangla License Plate Recognition Using Convolutional Neural Networks (CNN)." arXiv preprint arXiv:1809.00905 (2018).

[2] Abdullah, Sohaib, Md Mahedi Hasan, and Sheikh Muhammad Saiful Islam. "YOLO-Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City." In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1-6. IEEE, 2018.

[3] Baten, Raiyan Abdul, Zunaid Omair, and Urmita Sikder. "Bangla license plate reader for metropolitan cities of Bangladesh using template matching." In 8th International Conference on Electrical and Computer Engineering, pp. 776-779. IEEE, 2014.

[4] Dhar, Prashengit, Md Zainal Abedin, Razuan Karim, and Mohammad Shahadat Hossain. "Bangladeshi License Plate Recognition Using Adaboost Classifier In 2019 Joint 8th International Conference on Informatics,Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 342-347. IEEE, 2019.

[5] Saif, Nazmus, Nazir Ahmmed, Sayem Pasha, Md Saif Khan Shahrin, Md Mahmudul Hasan, Salekul Islam, and Abu Shafin Mohammad Mahdee Jameel. "Automatic License Plate Recognition System for Bangla License Plates using Convolutional Neural Network." In TENCON 2019-2019 IEEE Region 10 Conference (TENCON), pp. 925-930. IEEE, 2019.

[6] Haque, Md Rokibul, Saddam Hossain, Sagor Roy, Nashid Alam, and M. Jahirul Islam. "Line segmentation and orientation algorithm for automatic bengali license plate localization and recognition." International Journal of Computer Applications 154, no. 9 (2016).

[7] Ghosh, Ashim Kumar, Swapan KD Sharma, Md Nazrul Islam, Suchitra Biswas, and Sameya Akter. "Automatic license plate recognition (alpr) for bangladeshi vehicles." Global Journal of Computer Science and Technology (2011).

[8] Joarder, Md Mahbubul Alam, Khaled Mahmud, Ahmed Tasnuva, Mohsina Kawser, and Bulbul Ahamed. "Bangla automatic number plate recognition system using artificial neural network." (2012).

[9] Pias, Tanmoy Sarkar, David Eisenberg, and Muhammad Aminul Islam. "Vehicle Recognition Via Sensor Data From Smart Devices." In 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 96-99. IEEE, 2019.

[10] Hassan, Reshad, Sakib Hasan, Md Jubaer Hasan, Md Rafat Jamader, David Eisenberg, and Tanmoy Pias. "Human Attention Recognition with Machine Learning from Brain-EEG Signals." In 2020 IEEE 2nd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), pp. 16-19. IEEE, 2020.

[11] Rahman, Rashik, Tanmoy Sarkar Pias and Tanjina Helaly. "GGCS: A Greedy Graph-Based Character Segmentation System for Bangladeshi License Plate." in press.