International Journal of New Computer Architectures and their Applications (IJNCAA) 11(1): 1-9
The Society of Digital Information and Wireless Communications, 2021 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

# Segmentation-free Bangladeshi License Plate Recognition Using YOLO with Heuristic Bounding Box Refinement

Mahmudul Hasan Bhuiyan mahmudulcsecu@gmail.com
Muhammad Anwarul Azim azim@cu.ac.bd
Mohammad Khairul Islam mkislam@cu.ac.bd
Farah Jahan farah_csc@cu.ac.bd

Department of Computer Science and Engineering
University of Chittagong, Chittagongon-4331, Bangladesh

## ABSTRACT

Automatic License Plate Recognition is the core of Intelligent Transportation System because of its diverse set of applications. There are some online APIs such as OpenALPR, Sighthound, etc. available for recognizing license plates of some countries; however, no API is available for recognizing Bangladeshi license plates. The aforementioned APIs cannot even localize Bangladeshi license plates. In this paper, we employ the Fast YOLO detector for the first time to Bangladeshi license plate recognition. We also propose an aspect ratio-based bounding box refinement technique that experimentally outperforms the conventional way of fixed padding. Various performance evaluation metrics show that our proposed system not only provides impressive performance but also outperforms the state-of-the-art methods of license plate recognition. Additionally, we also introduce an annotated dataset containing samples of different challenging situations which will help to reduce the scarcity of benchmark datasets in this domain.

## KEYWORDS

Automatic License Plate Recognition, YOLO, Transfer Learning, Bounding Box Refinement.

## 1 INTRODUCTION

Automatic License Plate Recognition (ALPR) works as the heart of Intelligent Transportation System [1]. It has a lot of applications including automatic parking management, automatic toll collection, traffic rules violation detection and law enforcement, access and border control and so on [2]. A typical ALPR system consists of three major steps: (i) License plate localization; (ii) Character segmentation, and (iii) Character recognition. The license plate localization step is responsible for localizing the License Plate (LP) in a given image, character segmentation isolates individual characters from the detected LP and finally, character recognition aims at recognizing isolated characters. Because of the importance of successful localization of the LP, many approaches first detect the vehicle from an input image and then the LP of the vehicle to reduce false positive responses [3]. Despite its enormous real-life applications, ALPR is still one of the challenging problems in computer vision [4]. The difficulties of the problem lie in the highly complicated background, the variability of LPs, and random photographing conditions such as illumination, distortion, occlusion or blurring [5]. Due to these challenges, most of the works are done under some constraints such as specific distant cameras or viewing angles, good lighting conditions, simple backgrounds and considering certain regions or types of vehicles [3].

According to the statistics of Bangladesh Road Transport Authority (BRTA), a government agency, in March 2018 the number of registered vehicles in this country is over 3.4M. It is easy to realize that ALPR system is a must to maintain the traffic and ensuring accountability of traffic rules violation in such a highly populated country. To ease traffic management, the Bangladesh government has standardized the LP by making it compulsory to use Retro-Reflexive license plates for the ease of visualization in different light conditions. There

International Journal of New Computer Architectures and their Applications (IJNCAA) 11(1): 1-9
The Society of Digital Information and Wireless Communications, 2021 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

are some variations in the color and size of the LPs. A typical LP is shown in Figure 1.


Figure 1. Different parts of a typical Retro-Reflexive license plate used in Bangladesh.

The first row of a LP contains one of the 64 District Codes (DC) which specifies the district where the vehicle is originally registered, Metropolitan Code (MC) specifies whether the district is a metropolitan or not, and Vehicle Class Character (VCC) specifies type of the vehicle. There are 31 types of vehicle defined by the BRTA; for example 'ja' for public minibus, 'ga' for motor car of 1301 to 2000 Cubic Centimeters (CC) engine etc. If a vehicle is not registered in any metropolitan city, then the MC remains absent. A total of six digits in the second row specify the registration number of the vehicle.

## 2 RELATED WORKS

Bangladeshi vehicle images collected from certain distances and angles were considered in [6]. They used empirical techniques to locate the region of interests, horizontal and vertical projections for segmentation and neural network for recognition.

In [7], the authors applied Clip Limited Adaptive Histogram Enhancement (CLAHE) to process the input images of Bangladeshi LP before localizing. Character segmentation was done by using some character features such as pixel connectivity, line space and size etc. Several distinctive and mathematical features from the characters were used for character recognition. For isolating the character in LPs, a special feature of Bengali called "Matra" (horizontal line at the upper portion of a character) was used in [8]. Morphological operations were applied for extracting the LP and correcting the skew and rotation. The "Matra" was used to segment individual characters and finally, a neural network was used for recognition.

The conventional three stage pipeline of ALPR was employed in [9]. Morphological operations such as dilation and erosion were used for extracting candidate regions. The regions were binarized with adaptive thresholding and then Connected Component Analysis (CCA) was employed to segment the characters. Finally, Support Vector Machine (SVM) was used for classification.

In [10], an ALPR system that required to specify the shape and dimension of the LP was proposed. The authors compared different contour detection and object extraction methods. The possibility of using content based CCA for LP localization is discussed in [11].

LP localization in various hazardous conditions was considered in [2]. Different techniques such as noise removal, contrast enhancement, tilt correction, rain removal filter etc. were used to correctly localize the license plates.

In [4], the authors fine-tuned three deep Convolutional Neural Network (CNN) for localization, segmentation and recognition. They named the API as 'Sighthound' and showed that it outperformed several benchmark ALPR systems when evaluated on two benchmark datasets of USA and European License plates.

The paper in [3] proposed an ALPR system based on the state-of-the-art You Only Look Once (YOLO) object detection model. Their model performed better than 'OpenALPR' API and 'Sighthound' on SSIG dataset which is the largest public dataset of Brazilian LPs. They also introduced a more robust and richer public dataset for the ALPR task named 'UFPR-ALPR dataset'.

Chandra et al. [12] used CLAHE, Otsu's thresholding and Contour Analysis for Bangladeshi LP localization and projection for character segmentation. Google Tesseract Optical Character Recognition (OCR) Engine was used to recognize the MC and two other CNNs were trained to recognize the VCC and digits.

Morphological operations and CNN based ALPR system for recognizing Bangladeshi LPs was proposed by Rabbani et al. [13]. Size and aspect ratio of the LPs were used for LP localization and CCA is used for character segmentation. Finally, CNN was used for character recognition.

In [14], the authors presented a system to detect the Bangladeshi LP with white background only. They applied morphological analysis and shape verification using distance to border vectors (DtBs) for LP detection, and CNN for recognizing the segmented characters.

The authors of [1] employed YOLOv3 and ResNet-20 architecture for recognizing only Dhaka metropolitan vehicle LPs. They used YOLO for detecting LP and its digits and ResNet-20 architecture for the VCC. As only Dhaka metropolitan city vehicles were considered, they omit the part of DC and MC.

A BLPR system on video stream was presented in [15]. MobileNetV2 based CNN is used for vehicle detection from video frames. Then YOLOv4 object detection model is used for detecting the LP and Google Tesseract OCR engine is used for recognizing the characters. The dataset was created by capturing video of the traffic in Dhaka city.

A three step BLPR system was proposed in [16]. First, the LP is detected using YOLOv3, then a segmentation method based on Breadth First Search (BFS) and Flood-fill algorithm was applied and finally a CNN was used for character recognition. In their system, they only considered the LP of Dhaka metropolitan city.

A two-step deep CNN based approach is presented in [17]. They fine-tuned Faster R-CNN and SSD MobileNet for both the LP detection and Character recognition. They presented the performance of different combinations of the selected CNNs, however the size of the test data was very small.

# 3 METHODOLOGY

The classical methods have been used in Bangladeshi License Plate Recognition (BLPR) are restrictive and not robust. Based on our investigation, no such work was found on the whole BLPR pipeline using pre-trained CNN detectors. No benchmark dataset has been referenced in the literature for BLPR and hence researchers used their own datasets. In our work, we use the YOLO detector for the whole BLPR pipeline pre-trained on the PASCAL VOC dataset. We also propose a dataset [1] for the BLPR so that future works can show comparative performance on this dataset.

## 3.1 System Design

We propose a Detect Plate and Detect Characters design for the BLPR pipeline. This implies that we use one CNN for LP detection and another for character detection. This design provides the best compromise among processing speed, complexity, and accuracy. The segmentation is done directly by CNN as a subprocess of character detection. The absence of the segmentation step reduces the error propagation and results in a less overall system error. The Fast YOLOv2 $416 \times 416$ version is chosen for both the LP detection and character detection. It is fast enough for processing real-time video and provides high accuracy [18]. Both models can be fine-tuned separately to improve the performance of the corresponding step.

The proposed BLPR pipeline consists of two Fast YOLOv2 models named ‘Plate Detection CNN’ and ‘Character Detection CNN’. The plate detection CNN detects the LP from a given vehicle image which has been passed to the character detection CNN for detecting the characters. It detects all the contents of an LP, i.e., the DC, MC, VCC, and the registration number. The architecture of our system is shown in Figure 2.

## 3.2 Dataset Preparation

The experimental dataset has been formed by combining the dataset of [12] and the data collected by ourselves. It contains a total of 500 images with samples of Dhaka, Chattogram

[1] http://bit.ly/BLPR-Dataset

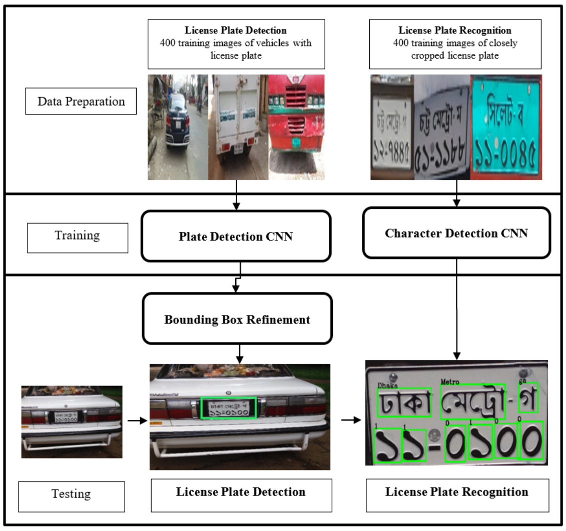


Figure 2. Architecture of the proposed system.

(a)

(b)

Figure 3. Few examples from the dataset. (a) 12 images from the training set and (b) 12 images from the test set.

and Sylhet divisions. We split the dataset into a training set of 400 and a test set of 100 images, every image contains a vehicle image with LP. The test set has been formed in such a way that it contains samples with different challenges such as slanted, blur, low light, shadow and night images, etc. and it contains number of samples of each of the cities in accordance with their participating portion in the training set. Some samples from the dataset are shown in Figure 3.

YOLOv2 supports annotation in PASCAL VOC format where an XML annotation file is generated for an image to represent the contents of the image by means of some fixed tags. All the images of the dataset are manually annotated. In the annotation of vehicle images, only one bounding box represents the whole license plate and in the annotation of closely cropped license plate images. Each of the bounding boxes represents any of the DC, MC, VCC or digit.

## 3.3 Training

As the number of data samples in our training set is not so high, fine-tuning the whole network is not feasible. Instead, we have modified the last two layers of CNNs and fine-tuned them with our data. Vehicle images have been used for training the plate detection CNN where the annotations give the bounding boxes of the LPs and the character detection CNN has been trained on closely cropped license plate images where bounding boxes represent the contents of the license plates. The Table 1 shows the modified architecture of the CNNs.

Table 1. Network architecture of the CNNs of our proposed approach.

| | | | Plate Detection CNN | | Character Detection CNN | |
|---|---|---|---|---|---|---|
| Layer | Layer Type | Size/Stride | Filters | Output | Filters | Output |
| 0 | Input | | | $416 \times 416 \times 3$ | | $416 \times 416 \times 3$ |
| 1 | Convolutional | $3 \times 3$ | 16 | $416 \times 416 \times 16$ | 16 | $416 \times 416 \times 16$ |
| 2 | Maxpool | $2 \times 2/2$ | | $208 \times 208 \times 16$ | | $208 \times 208 \times 16$ |
| 3 | Convolutional | $3 \times 3$ | 32 | $208 \times 208 \times 32$ | 32 | $208 \times 208 \times 32$ |
| 4 | Maxpool | $2 \times 2/2$ | | $104 \times 104 \times 32$ | | $104 \times 104 \times 32$ |
| 5 | Convolutional | $3 \times 3$ | 64 | $104 \times 104 \times 64$ | 64 | $104 \times 104 \times 64$ |
| 6 | Maxpool | $2 \times 2/2$ | | $52 \times 52 \times 64$ | | $52 \times 52 \times 64$ |
| 7 | Convolutional | $3 \times 3$ | 128 | $52 \times 52 \times 128$ | 128 | $52 \times 52 \times 128$ |
| 8 | Maxpool | $2 \times 2/2$ | | $26 \times 26 \times 128$ | | $26 \times 26 \times 128$ |
| 9 | Convolutional | $3 \times 3$ | 256 | $26 \times 26 \times 256$ | 256 | $26 \times 26 \times 256$ |
| 10 | Maxpool | $2 \times 2/2$ | | $13 \times 13 \times 256$ | | $13 \times 13 \times 256$ |
| 11 | Convolutional | $3 \times 3$ | 512 | $13 \times 13 \times 512$ | 512 | $13 \times 13 \times 512$ |
| 12 | Maxpool | $2 \times 2/1$ | | $13 \times 13 \times 512$ | | $13 \times 13 \times 512$ |
| 13 | Convolutional | $3 \times 3$ | 1024 | $13 \times 13 \times 1024$ | 1024 | $13 \times 13 \times 1024$ |
| 14 | Convolutional | $3 \times 3$ | 1024 | $13 \times 13 \times 1024$ | 1024 | $13 \times 13 \times 1024$ |
| 15 | Convolutional | $1 \times 1$ | 30 | $13 \times 13 \times 30$ | 165 | $13 \times 13 \times 165$ |
| 16 | Softmax | | | | | |

To facilitate the training and prediction process, each of the possible Bengali objects present in a LP has been mapped to their corresponding English counterparts. The Table 2 shows the mapping of Bengali objects to their respective English counterparts.

Table 2. Mapping of Bengali objects to their English counterparts.

| Bengali | English | Bengali | English | Bengali | English |
|---|---|---|---|---|---|
| ঢাকা | Dhaka | জ | ja | ০ | 0 |
| চট্ট | Chatta | ঝ | jha | ১ | 1 |
| সিলেট | Sylhet | ঠ | tha | ২ | 2 |
| মেট্রো | Metro | ন | na | ৩ | 3 |
| ক | ka | ব | ba | ৪ | 4 |
| খ | kha | ভ | bha | ৫ | 5 |
| গ | ga | ম | ma | ৬ | 6 |
| ঘ | gha | স | sa | ৭ | 7 |
| চ | ca | – | – | ৮ | 8 |
| ছ | cha | – | – | ৯ | 9 |

The training of the plate detection network is stopped after 50 epochs with an error of 0.96 to avoid over-fitting. Whereas, the character detection network has been trained for 100 epochs and the loss remain as high as 5.19. The reason behind the high loss of the network might be the high number of character classes and the rareness of most of them in the training samples. The learning curve of the networks are shown in Figure 4.

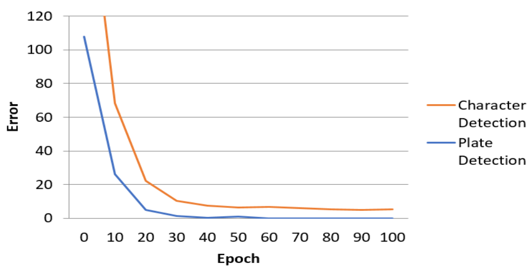


Figure 4. Learning curves of the CNNs.

### 3.4 Choice of Confidence Thresholds

The confidence threshold value has been used to control the false positive and false negative rate in object detection. The prime concern of the plate detection network is to detect all the license plates as there is no backup system to detect the undetected plates. The only way to find an undetected license plate is to manually examining the result of detection which is contrary to the aim of the system. If the threshold value is set low and a false license plate is detected, it does not affect the system performance significantly because the character detection network can't detect any character from that. Contrarily, for the character detection network, we aim to detect only the characters on which the network is fairly confident. If the character detection network makes a false positive prediction, then there is no way to discover the error without manually checking all the predictions which is also contradictory to the aim of the system. After testing with different confidence threshold values, the confidence threshold value for plate detection CNN has been set to 0.2 and the value for character detection CNN has been set to 0.25.

### 3.5 Bounding Box Refinement

Our LP detector attains an average Intersection over Union (IoU) of 0.7298, however, it still cutdowns some characters. The aspect ratio of an object remains the same as the distance and the size of an image changes unless there is a change in viewpoint. It has a close relationship with the IoU of the predicted bounding box. Our experiments have shown that the aspect ratio of bounding boxes with high IoU ranges roughly from 1.7 to 2.3. If the aspect ratio is less than 1.7, in most cases width of the bounding box is smaller in comparison to the height and if the aspect ratio is greater than 2.3, the width of the bounding box is larger in comparison to the height. Therefore, by taking these values as thresholds, we have found the best combination of margin for each of the cases as shown in Figure 5. This aspect ratio based bounding box refinement improves localization accuracy in place of adding a fixed margin.

## 4 EXPERIMENTAL RESULT

We have performed our experiment using an Intel Core i5 7th generation 2.5GHz computer with 8GB RAM. Our system has been implemented in Python 3 on a Windows platform. The required APIs are TensorFlow, OpenCV 3.0 and Darknet which is an open-source neural network framework.

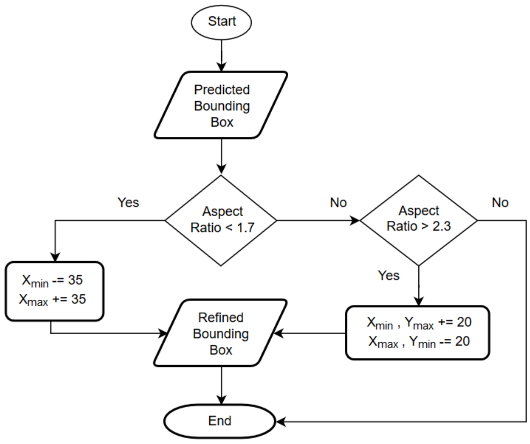


Figure 5. Flow chart of bounding box refinement.

(a)


(b)

Figure 6. Results of plate detection in some challenging samples. (a) FN detections and (b) TP detections.

## 4.1 Performance of Bounding Box Refinement

Application of our bounding box refinement has increased the average IoU by 0.0254 i.e., from 0.7298 to 0.7552. Whereas, the maximum average IoU obtained is 0.7396 with a fixed margin of 20 pixels. This increased IoU contributes in obtaining better recognition accuracy by reducing cutout objects as pointed out in Table 3.

Table 3. Performance improvement of license plate detection with bounding box refinement.

| Metric | Before | After |
|---|---|---|
| Average IoU | 0.7298 | 0.7552 |
| No. of cutout objects | 17 | 11 |

## 4.2 Performance of Plate Detection CNN

The performance of plate detection CNN has been evaluated on the test set of the vehicle images. The mean Average Precision (mAP) gives the average precision for different object classes. The obtained mAP for plate detection is 94.95% and it is the same as the average precision in this case as we have only one class. Figure 6 shows some True Positive (TP) and False Negative (FN) detection results.

## 4.3 Performance of Character Detection

The performance of the character detection CNN has been evaluated on the test set of closely cropped license plates. The Figure 7 shows the number of True Positive (TP) and False Positive (FP) for different object classes.

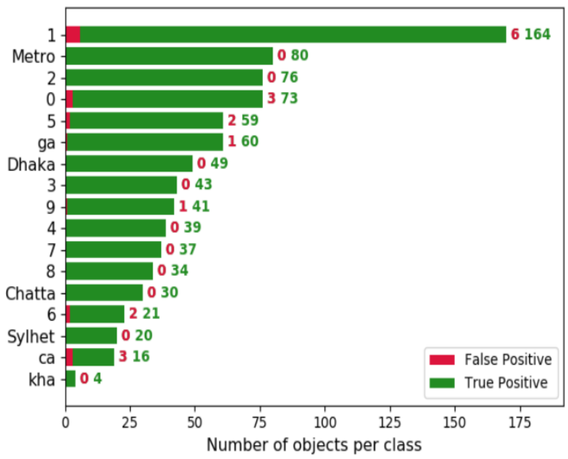


Figure 7. Number of TP and FP samples for each object class.

The mAP along with average precision values for different object classes are shown in Figure 8. The mAP of our character detection network is 64.21%. Since the mAP is the arithmetic mean of the average precision for different classes, it is adversely affected by the lowest precision obtained for the rare classes.

Our character detection network cannot recognize some vehicle class characters which are very rare (< 15 samples) in the training dataset. The mAP is greatly affected by these

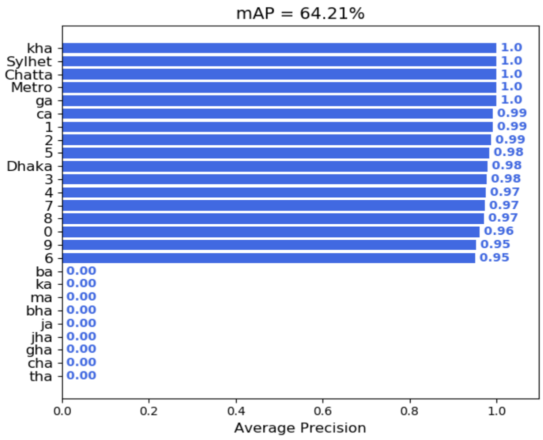


Figure 8. The mAP of character detection CNN.

classes. If only the classes detected by the network are considered for calculating the mAP value, then the mAP value will be as high as 98.12% which is much higher than the overall mAP value of 64.21%. Figure 9 shows correct of characters by our method in some challenging samples.

Figure 9. Examples of character detection for some challenging samples.

### 4.4 Performance Comparison

Table 4 compares the detection performance of our system and the state-of-the-art method proposed in [12].

Table 4. Comparison of license plate detection accuracy of our proposed method and Chandra et al. [12].

| System | Correctly Detected | Accuracy (%) |
|---|---|---|
| Chandra et al. [12] | 64 | 64 |
| Our System | 95 | 95 |

The authors of [12] used Google's Tesseract OCR for recognizing the DC and MC. Our character detection CNN itself can do the same job. However, we only compare the performance of VCC and digits recognition of both systems in Table 5.

Table 5. Comparison of character and digit recognition accuracy (in %) of our proposed method and [12].

| System | Character Recognition | | | Digit Recognition | | |
|---|---|---|---|---|---|---|
| | Total | Success | Accuracy | Total | Success | Accuracy |
| Chandra et al. [12] | 64 | 50 | 78.13 | 384 | 328 | 85.42 |
| Our System | 95 | 80 | 84.21 | 570 | 556 | 97.55 |

Run time is an important factor in developing real time ALPR systems. The comparison of the running time of our approach and that presented in [12] is given in the Table 6.

Table 6. Comparison of running time of our proposed method and [12].

| System | Total time (s) | Time per image (ms) |
|---|---|---|
| Chandra et al. [12] | 21.62 | 216.2 |
| Our System | 54.43 | 544.3 |

The deep CNNs are specially developed for running on GPUs. Different experiments show that running on GPU can speed up CNN processing by a factor of 10 or more based on the type of GPU used [19]. Our system is expected to provide real time performance when run on a GPU like the original YOLO. However, our current system is slower than the system used in [12] when run on CPU.

## 5 CONCLUSION AND FUTURE WORK

This paper presents a Fast YOLO based Bangladeshi License Plate Recognition system with an aspect ratio based bounding box refinement technique to improve the detection of the original YOLO network. Two pretrained CNNs are incorporated for license plate localization and recognition where the last two layers of them are fine-tuned for transfer learning the general features. Experimental results show that the proposed approach outperforms the system proposed in [12] in terms of accuracy and also demonstrates the capability of meeting the robustness requirement of the real-world operating environments. A PASCAL VOC format annotated dataset is also presented to address the scarcity of benchmark

datasets in this domain. Two separate networks are trained and fine-tuned using vehicle images and carefully cropped license plate images for license plate localization and license plate recognition respectively.
We aim to improve our research findings by exploiting the latest YOLO models and other CNN models such as variants of R-CNN which are well suited for directly reading text from an image. Other possibilities of improving bounding box refinement using license plate features such as edge and border can also be investigated. The challenge posed by a high variety of rare vehicle class characters and scarcity of real image can be addressed by carefully fabricating artificial license plate images.

## REFERENCES

[1] S. Abdullah, M. Mahedi Hasan, and S. 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)*, pages 1–6, 2018.

[2] Samiul Azam and Md Monirul Islam. Automatic license plate detection in hazardous condition. *Journal of Visual Communication and Image Representation*, 36:172–186, 2016.

[3] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2018.

[4] Syed Zain Masood, Guang Shu, Afshin Dehghan, and Enrique G Ortiz. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*, 2017.

[5] H Li, P Wang, and C Shen. Towards end-to-end car license plates detection and recognition with deep neural networks. corr abs/1709.08828 (2017).

[6] M Shadab Mashuk, M Arif Majid, Naimul Basher, and Tarif Riyad Rahman. Automatic detection of bangla characters in bangladeshi car registration plates. In *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, pages 166–171. IEEE, 2010.

[7] Nahian Alam Siddique, Asif Iqbal, Fahim Mahmud, and Md Saifur Rahman. Development of an automatic vehicle license plate detection and recognition system for bangladesh. In *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*, pages 688–693. IEEE, 2012.

[8] Raiyan Abdul Baten, 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*, pages 776–779. IEEE, 2014.

[9] Md Azher Uddin, Joolekha Bibi Joolee, and Shayhan Ameen 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.

[10] Charbel Fares. Intelligent license plate recognition. *Inter. J. New Comp. Architectures Applic*, 3:54–69, 2013.

[11] Ismail A El Sayad, Mohammad A Bazzoun, Hawraa I Younes, Laila M Ghoteime, and Samih Abdulnabi. Content based image retrieval using multimodal data based on cca. *International Journal of New Computer Architectures*

*and Their Applications*, 8(3):119–125, 2018.

[12] Sudipto Chandra, Md Al-amin Nowshad, M Jahirul Islam, et al. An automated system to detect and recognize vehicle license plates of bangladesh. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6. IEEE, 2017.

[13] Golam Rabbani, Mohammad Aminul Islam, Muhammad Anwarul Azim, Mohammad Khairul Islam, and Md Mostafizur Rahman. Bangladeshi license plate detection and recognition with morphological operation and convolution neural network. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–5. IEEE, 2018.

[14] Prashengit Dhar, Sunanda Guha, Tonoy Biswas, and Md Zainal Abedin. A system design for license plate recognition by using edge detection and convolution neural network. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pages 1–4. IEEE, 2018.

[15] Md Onim, Saif Hassan, Muhaiminul Islam Akash, Mahmudul Haque, and Raiyan Ibne Hafiz. Traffic surveillance using vehicle license plate detection and recognition in bangladesh. *arXiv preprint arXiv:2012.02218*, 2020.

[16] Md Mesbah Sarif, Tanmoy Sarkar Pias, Tanjina Helaly, Md Sohel Rana Tutul, and Md Nymur Rahman. Deep learning-based bangladeshi license plate recognition system. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6. IEEE, 2020.

[17] Tariqul Islam and Risul Islam Rasel. Real-time bangla license plate recognition system using faster r-cnn and ssd: A deep learning application. In *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, pages 108–111. IEEE.

[18] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[19] Saide Isilay Baykal, Deniz Bulut, and Ozgur Koray Sahingoz. Comparing deep learning performance on bigdata by using cpus and gpus. In *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, pages 1–6. IEEE, 2018.