**Department of Electrical and Computer Engineering**

**North South University**



**Intern/Co-op/Directed Research**

**Bengali License Plate Recognition System Using Deep Learning**

**Nawal Ayesha Khan ID# 1911301042**

**Sarah Jasim ID# 1912260042**

**Seam E Nur Sami ID# 1912489042**

**Faculty Advisor:**

**Dr. Riasat Khan**

**Assistant Professor**

**ECE Department**

**Fall 2022**

**APPROVAL**

Nawal Ayesha Khan (ID # 1911301042), Sarah Jasim (ID # 1912260042) and Seam-E-Nur Sami (ID # 1912489042) from Electrical and Computer Engineering Department of North South University, have worked on the CSE498R Project titled “Bengali License Plate Recognition System Using Deep Learning” under the supervision of Dr. Riasat Khan to partially fulfill the requirements for the degree of Bachelor of Science in Computer Science and Engineering on March 5th, 2023, and has been accepted as satisfactory.

**Supervisor’s Signature**

…………………………………….

**Dr. Riasat Khan**

**Assistant Professor**

Department of Electrical and Computer Engineering

North South University

**Chairman’s Signature**

…………………………………….

**Dr. Rajesh Palit**

**Professor and Chair**

Department of Electrical and Computer Engineering

North South University

**DECLARATION**

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students’ Names & Signatures

**1. Nawal Ayesha Khan**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**2. Sarah Jasim**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**3. Seam E Nur Sami**

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**ACKNOWLEDGEMENTS**

Firstly, we would like to thank our course instructor, Riasat Khan, for his direction, management, and coordinated effort in helping us accomplish our Intern/Co-op/Directed Research with the dedication, focus, enthusiasm, and resolution needed. We appreciate his ceaseless help, supervision, advice, and recommendation that allowed us to finish this report.

Our sincere thanks also goes to North South University, Dhaka, Bangladesh for giving us such a platform where we can have an industrial level experience as a part of our academics.

Last but not least, we would like to thank our family as their inspiration and guidance kept us focused and motivated.

**ABSTRACT**

Bengali License Plate Recognition System Using Deep Learning

Automatic License Plate Recognition systems are a frequent research topic worldwide. They play a significant role in many applications, such as road traffic monitoring, law enforcement, automatic parking systems, etc. Traffic violations and crime are prevalent in Bangladesh, so there is an urgent need for a suitable ALPR system. This paper presents a deep-learning approach for license plate detection and recognition in Bangladesh. Our detection system combines the latest object detection algorithm, YOLOv8, with EasyOCR.

The dataset for this paper has been compiled from multiple Kaggle datasets, a Bangla license plate database, and photographs randomly taken of vehicles on the road. For preprocessing the dataset, Roboflow has been used to build the dataset and mark bounding boxes for the plates. Our proposed architecture consists of two stages. In the first stage, we use YOLOv8 to detect or localize license plates, and in the second stage, we utilize the EasyOCR package to recognize Bengali characters. Our YOLOv8 model achieved a mean average precision value of 0.964, while the EasyOCR reader performed well in the character recognition segment. The experimental results show that the proposed system can accurately localize and detect Bengali characters on license plates.

TABLE OF CONTENTS

[1.1 Introduction 2](#_Toc128992152)

[Chapter 2 5](#_Toc128992153)

[Literature Review 5](#_Toc128992154)

[2.1 Related Works 6](#_Toc128992156)

[Chapter 3 10](#_Toc128992157)

[Methodology 10](#_Toc128992158)

[3.1 Introduction 11](#_Toc128992160)

[3.2 Dataset and Pre-processing 11](#_Toc128992161)

[3.3 Detection and Recognition Stages 11](#_Toc128992162)

[Chapter 4 15](#_Toc128992163)

[Results and Discussion 15](#_Toc128992164)

[4.1 Results of YOLOv8 model simulations 16](#_Toc128992166)

[4.2 Results of EasyOCR 20](#_Toc128992167)

[Chapter 5 24](#_Toc128992168)

[Conclusion and Future Work 24](#_Toc128992169)

[5.1 Conclusion 25](#_Toc128992171)

[Bibliography 26](#_Toc128992172)

**LIST OF FIGURES**

[Fig. 1. The standard BRTA license plate format 3](#_Toc36064369)

[Fig. 2. Architecture of YOLOv8 model 12](#_Toc36064370)

[Fig. 3. Working sequence of our proposed system 14](#_Toc36064374)

[Fig. 4. mAP50 curve (Left) and mAP50-95 curve (Right) for the license plate detection. 16](#_Toc36064377)

[Fig. 5. Precision curve (Left) and Recall curve (Right) for license plate detection 16](#_Toc36064383)

[Fig. 6. Box loss curve (Left), Classification loss curve (Middle), and Recall curve (Right) for license plate detection. 17](#_Toc36064384)

[Fig. 7. Confusion Matrix between classes (license\_plate and background) 1](#_Toc36064384)7

[Fig. 8. Precision-Confidence curve for the license plate detection 18](#_Toc36064384)

[Fig. 9. Recall-Confidence curve for the license plate detection 18](#_Toc36064384)

[Fig. 10. Precision-Recall curve for the license plate detection 18](#_Toc36064384)

[Fig. 11. F1-Confidence curve for the license plate detection. 19](#_Toc36064384)

[Fig. 11. F1-Confidence curve for the license plate detection. 19](#_Toc36064384)

[Fig. 12: Model predictions on validation images (1) 19](#_Toc36064384)

[Fig. 13. Model predictions on validation images (2) 20](#_Toc36064384)

[Fig. 14. EasyOCR results for “ঢাকা মেট্রো-গ” “৩৯-১৪৭৩” 21](#_Toc36064384)

[Fig. 15. EasyOCR Results for “চট্ট মেট্রো-গ” “১২-৬৪১৭” 21](#_Toc36064384)

[Fig. 16. EasyOCR Results for “ঢাকা মেট্রো-গ” “৩৯-১৭৮৬” 21](#_Toc36064384)

**LIST OF TABLES**

[Table I. HYPER-PARAMETER SETTINGS FOR YOLOV8 MODEL 13](#_Toc36064834)

[Table II. Comparison table of model performance 22](#_Toc36064834)

**Chapter 1**

**Introduction**

## Introduction

Automatic License Plate Recognition (ALPR) or Vehicle License Plate Detection (VLPD) has been a frequent research topic and plays a vital role in many applications, such as road traffic monitoring, law enforcement, automatic parking systems, toll collection, access, border control, etc. It is the heart of intelligent transportation systems around the world. ALPR systems are also crucial to Bangladesh due to its high rate of traffic crimes and road accidents. According to Bangladesh Road Transport Authority (BRTA), the total number of registered motor vehicles in 2022 was reported at 578,151.000 units [1]. However, numerous unregistered and unfit vehicles continue to operate on the roads. The BRTA found the number of vehicles without fitness certificates to be 5.08 lakh as of January 2022 [19]. The number of unfit vehicles across the country has been on the rise over the years, often causing fatal road accidents. Traffic law violations, crime, and incessant traffic jams are rampant in this country, and hence, the need for a suitable ALPR system is now even greater than ever.

In Bangladesh, the BRTA issues registration plates for each vehicle. According to its regulations, all license plates are written in Bengali with a fixed two-line text format and color for different types of vehicles. The general format of Bangladeshi vehicle license plates is "City Metro - Vehicle Class - Vehicle Number", for example: "DHAKA METRO-Ga-12-3456". The "DHAKA" field represents the name of the district (vehicle registration area), followed by the word Metro (if the vehicle has been registered in a metropolitan area), and the "Ga" field represents the vehicle class. On the second line, six numerals are assigned to identify the vehicle. The "12" field represents the vehicle class in Bengali numerals, while the "3456" field after the hyphen represents the unique vehicle number. Fig. 1 illustrates the standard BRTA format. The colors of vehicle plates also hold some significance. For example, white license plates represent Private Service vehicles, while green license plates represent Public Service vehicles. License plates are generally installed on both the front and rear ends of the vehicle, the latter being a permanent attachment for the plate to the vehicle [20].

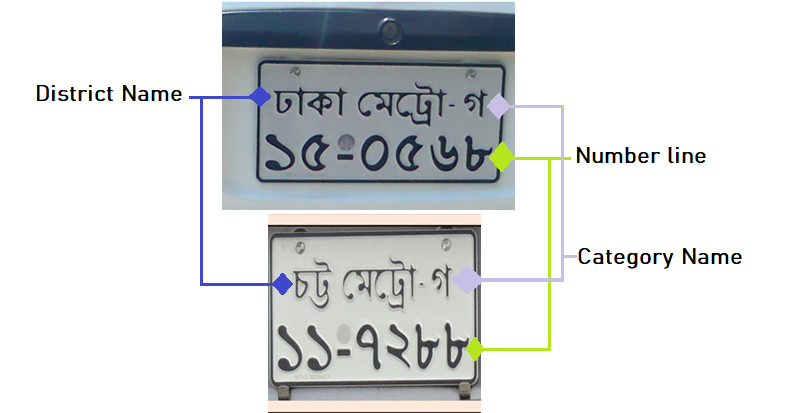
****

Fig. 1. The standard BRTA license plate format.

In most datasets, the license plate images have been captured under ideal conditions, such as proper lighting and a clear view of the license plate. However, in bustling cities like Dhaka or Chittagong, it is less likely for images of such situations to be collected. Datasets containing Bangla license plates representing real-life, non-ideal conditions, namely the presence of different lighting conditions, viewing angles, transparency, occlusion, etc., were unavailable. For this reason, we needed to compile a dataset representing these conditions and apply various levels of data augmentation to more accurately represent those non-ideal situations. We have also included plates from other cities in our dataset, in addition to the plates from Dhaka. These plates include those from Khulna, Chittagong, and a few more, allowing for greater variety in our data.

In this paper, we present a deep-learning based approach for a Bangla license plate recognition system, consisting of a license plate detection system followed by a Bengali character recognition system. We introduce a dataset by compiling 1,096 Bangladeshi vehicular license plate images collected from various online datasets resembling real-world scenarios. In this regard, we have employed the latest YOLOv8 algorithm to extract the license plate successfully. We have also utilized the EasyOCR package to recognize the Bengali alphabet and numerals. We have evaluated the performance of our model using standard metrics such as precision, recall, loss curves, etc. In our proposed method, the YOLOv8 model achieved 96.4% mean average precision (mAP), 96.5% precision, and 95.3% recall in license plate detection. At the same time, EasyOCR could also recognize the Bangla characters on the license plate adequately.Project Description

# 

# **Chapter 2**

# **Literature Review**



## Related Works

This section briefly reviews several recent papers that have used deep-learning approaches to design ALPR systems that recognize license plates of various languages, including Bangla.

Many papers have used YOLO models for their license plate detection stage and a CNN for character recognition. In [3], an effective ALPR system based on the YOLO object detector and CNN was trained and adjusted for each stage to function under various circumstances (e.g., variations in camera, lighting, and background). In this paper, the authors introduce a large, publicly available dataset called the UFPR-ALPR dataset, which consists of 150 videos and 4,500 frames of Brazilian license plates that were shot while both the camera and various automobiles were moving. The results they obtained from the two datasets using the proposed method were significant.

Similarly, [4] compared the effectiveness of YOLOv2 and YOLOv3 models in detecting and recognizing Chinese car license plates in complicated backgrounds. After 4,000 training steps, both YOLOv2 and YOLOv3 had recall and precision scores exceeding 99% on the test dataset. Their findings indicate that while YOLOv3 can outperform YOLOv2 in detection accuracy, the latter performs comparatively better in detection speed. The network CRNN-12, which combines the advantages of CNN and RNN, is employed in the recognition phase to identify license plates with different amounts of characters. The CRNN-12's maximum test accuracy was 98.86% (9873/9987) at 60,000 steps. They also built a custom dataset from pictures of varying conditions throughout the day and at multiple angles, numerous clarity levels, and various lighting situations.

Another YOLO-based Bengali license plate detection network was designed in [5], where they successfully localized the plates and recognized the numbers using the YOLOv3 algorithm. Many previous studies on Bangla license plates employed photographs taken under perfect circumstances for the ALPR because datasets depicting less-than-ideal circumstances were not readily available. To get around this, the authors have created a dataset of Bangla letters and numerals with more than 6,400 characters and used it to train a ResNet-20-based deep CNN. The proposed approach obtained an Intersection over Union (IoU) value greater than 85% in digit recognition. In contrast, the CNN model built on ResNet-20 recognized the Bangla letters on the license plate with 92.7% accuracy. The performance of the license plate detection on plates from other districts was not evaluated; instead, it was only evaluated on a dataset of Bangla license plates from the Dhaka metropolitan area. Despite having a few instances of failure, the suggested approach categorized numbers and characters with high accuracy under various circumstances.

In [6], they designed an intelligent vehicle license plate recognition system using still photos of Bangladeshi vehicles as input. First, image enhancement is performed with contrast stretching, followed by the Sobel operator to detect the edges of the plates. After character segmentation, feature extraction is performed, and finally, neural network techniques are used to recognize Bengali characters. They have also collected two sets of still photos for their dataset. However, the system has multiple limitations, such as poor performance due to fuzzy, blurry, and very distant images of the vehicles; characters being inaccurately classified for images taken in rainy or foggy situations; and similar characters, such as "0" and "O", "1" and "I" being incorrectly identified.

The main contribution of [8] was to determine the best four-layered CNN architectures that can be used as the algorithm in the character recognition task. The authors detected and extracted the license plates, after which they used connected component analysis to segment the characters before feeding them to the CNNs. They built their dataset by randomly capturing photos of vehicle license plates around the Malacca area in Malaysia. The preprocessing stage scored 74.7% in accuracy, while the CNN achieved 94.6% accuracy in recognizing the characters. However, this method also had some shortcomings, as the preprocessing stage needed further improvements to effectively remove noise, such as improper lighting conditions, and achieve higher accuracy.

In [2], they designed a real-time Bangladeshi license plate detection system which acquires input images of vehicles from videos captured by a Raspberry Pi camera and then performs plate detection and character recognition. Computer Vision approach and YOLOv3 algorithm were used for the license plate detection part, and Tesseract OCR and CNN were used for the character recognition part. They have also performed a comparative analysis of the performance of their deployed methods for each part. This analysis showed that the YOLOv3 algorithm performed better than the computer vision approach for plate detection, and CNN outperformed Tesseract OCR for the character recognition part. The YOLOv3 model had an accuracy of 95%, whereas the computer vision method had an accuracy of 91%. Tesseract OCR fed with a trained LSTM model on Bengali characters had an accuracy of 90%, while CNN had an accuracy of 91.38% for character recognition.

In [18], the proposed Bangladeshi VLPR system used a template matching method to detect license plates from still images of vehicles, after which it used the spatial super-resolution technique to enhance the visibility of low-resolution license plates. Character segmentation was then performed on the plates by drawing bounding boxes around the characters, and finally, character recognition was done by using CNN, which was trained using the AlexNet model. The template matching technique had a recognition rate of 86.5%, and the character recognition rate was 90.9%.

In [7], the authors compiled a standard, public dataset containing 1000 images and 79 video clips of Bangladeshi license plates under various conditions. Real-time license plate detection and recognition were then done using frames from the video clips. A Haar-cascade classifier initially detects the presence of clearly visible license plates within the frame, and then a more robust MobileNet SSDv2 model detects and extracts the plates from the filtered images. The cropped plates and their corresponding confidence values are then stored in a dictionary, and plates with the highest confidence value are finally passed onto a Vision API recognition module for optical character recognition (OCR), which returns the characters as a string. Using Haar-cascade in combination with MobileNet SSDv2 gave optimal results for real-time detection, as a very high FPS rate (27.1) could be achieved, as well as an AP0.5 score of 86.2%, precision of 63.6%, recall of 59.3% and an F1 score of 60.8% could be obtained.

# **Chapter 3**

# **Methodology**



## Introduction

For license plate detection, we have used the newest version of YOLO (You Only Look Once), the YOLOv8, while the Bengali character recognition is done by using the EasyOCR package. As detailed below, we have also used the Roboflow tool to build the dataset and mark bounding boxes for the license plates in the preprocessing stage.

## Dataset and Pre-processing

The dataset used for the plate detection portion has been compiled from multiple Kaggle datasets [9, 10, 11, 12]; the publicly available dataset of Bangladeshi vehicle images in [13], with visible Bangla license plates of almost all types of vehicles present in Bangladesh; as well as some collected from our cameras of vehicles on the road. The still images in this dataset are in JPG and PNG format. Although the total number of images after compiling the datasets amounts to 4,369 files, many files were unusable due to extreme blurriness, overly pixelated images, or even the lack of a plate or vehicle in the image. Due to these obstacles, we have included only relevant images for our study, leading to 1,096 images as the initial dataset. As YOLOv8 performs data augmentation on the images during its training online, the model would see a slightly different variation of the images provided at each epoch. The data augmentation eventually resulted in our dataset ultimately containing 3,840 images in total.

## Detection and Recognition Stages

1. YOLOv8: The YOLO (You Only Look Once) series of models has become famous due to its considerable accuracy while maintaining its small model size. YOLO models can be trained on a single GPU, making them accessible to a broader range of developers than most other detection models [14].

YOLOv8 is the latest YOLO model and can be used in tasks involving image classification, object detection, and instance segmentation. Developed by Ultralytics and launched on January 10th, 2023, it includes numerous architectural and developer changes and improvements over its well-known predecessor YOLOv5. It has a high accuracy rate, as measured by COCO and Roboflow 100, and comes with many developer-convenience features, such as its well-structured Python package. The architecture used in YOLOv8 has been detailed below in Fig. 2.

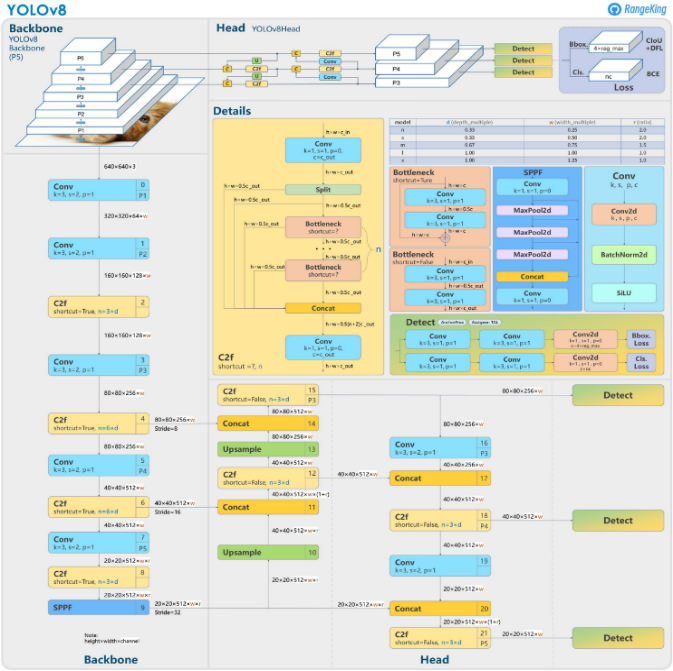


Fig. 2. Architecture of YOLOv8 model.

We originally intended to use YOLOv7 for this project but have since used the updated model to utilize YOLOv8’s improved features.

The model was trained using Google Colab, a web IDE that provides free access to powerful GPUs without conﬁguration. After preparing the license plate dataset, we trained the model by taking the initial weights from the pre-trained YOLOv8 model. The plate dataset was then split into 70% for the training set, 19% for the validation set, and 11% for the testing set. The hyper-parameters used for the YOLOv8 model have been detailed below in Table I.

TABLE I. HYPER-PARAMETER SETTINGS FOR YOLOV8 MODEL

|  |  |
| --- | --- |
| **Hyper-parameter** | **Value** |
| Optimizer | SGD (Stochastic Gradient Descent) |
| Epochs | 300 |
| Learning rate | 0.01 |
| Batch size | 16 |

1. EasyOCR: Based on a template matching algorithm, EasyOCR is a printed character reader that can read short texts (such as serial numbers, expiry dates, manufacturing dates, etc.) that have been printed or are directly on parts. It is a Python package that holds PyTorch as a backend handler and was created by the Jaided AI Company. Currently, it can detect more than 42 languages [16].

EasyOCR requires the font to be recognized. It can learn all possible characters that can be read from provided sample images, making recognition extremely flexible, fast, and reliable [17].

For the character recognition portion, we first extracted license plates from a sample of images from the test set. The EasyOCR reader is then initialized to Bengali, after which it is used to read the loaded images and read the text using a ‘greedy’ decoder. The reader then returns the characters detected with a confidence value.

A flowchart for the proposed working sequence has been displayed in Fig. 3.

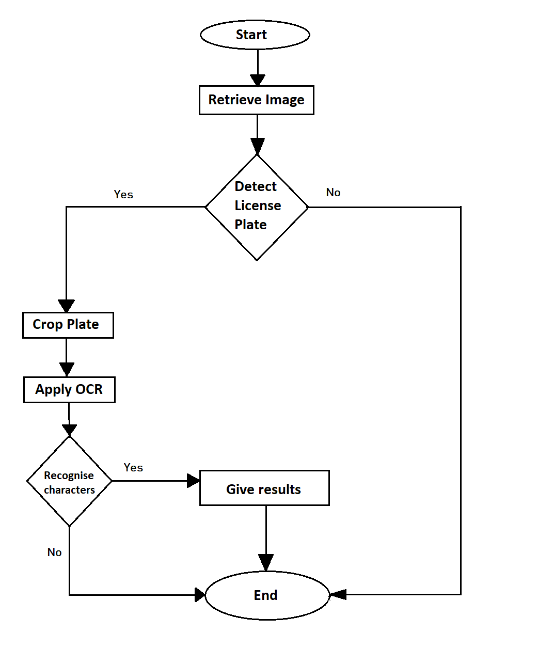


Fig. 3. Working sequence of our proposed system.

# **Chapter 4**

# **Results and Discussion**



## Results of YOLOv8 model simulations

The improvement in our model can be seen in the graphs from Fig. 4 to Fig. 6, which display the aforementioned performance metrics when the model makes predictions on the validation sets.

Fig. 4 shows the mean average precision values at IoU thresholds of 0.50 and 0.95. We can see that after 300 epochs, the mAP50 value levels off to 0.964, while the mAP50-95 value peaks at 0.718.

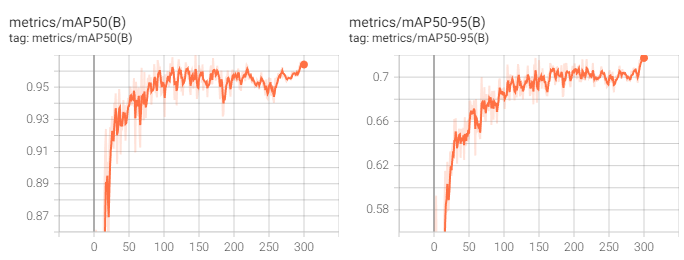
**

Fig. 4. mAP50 curve (Left) and mAP50-95 curve (Right) for the license plate detection.

The Precision and Recall curves in Fig. 5 show that after training, precision rises to 0.965, while recall increases gradually to 0.953.

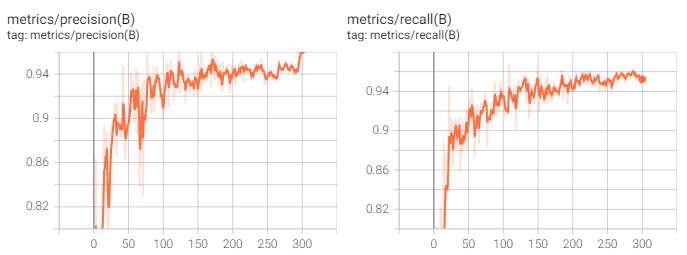
**

Fig. 5. Precision curve (Left) and Recall curve (Right) for license plate detection.

There are three different types of loss shown in Fig. 6: box loss, DFL loss, and classiﬁcation loss. The curves show that during training, the box loss of the model steadily drops to 0.540, while the classification loss has a sharp decline to 0.285, and the DFL loss decreases to 0.285.

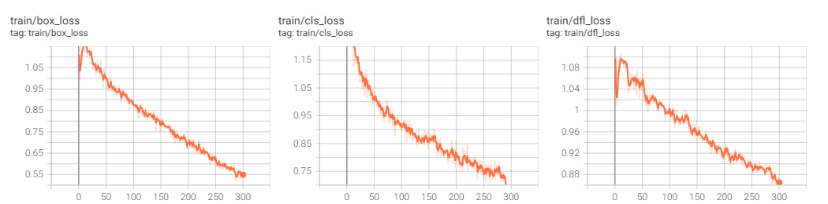
**

Fig. 6. Box loss curve (Left), Classification loss curve (Middle), and Recall curve (Right) for license plate detection.

The confusion matrix for the YOLOv8 model has been plotted in Fig. 7. From the results, we can see that the model correctly predicted 96% of the license plates from the images provided in the testing dataset.

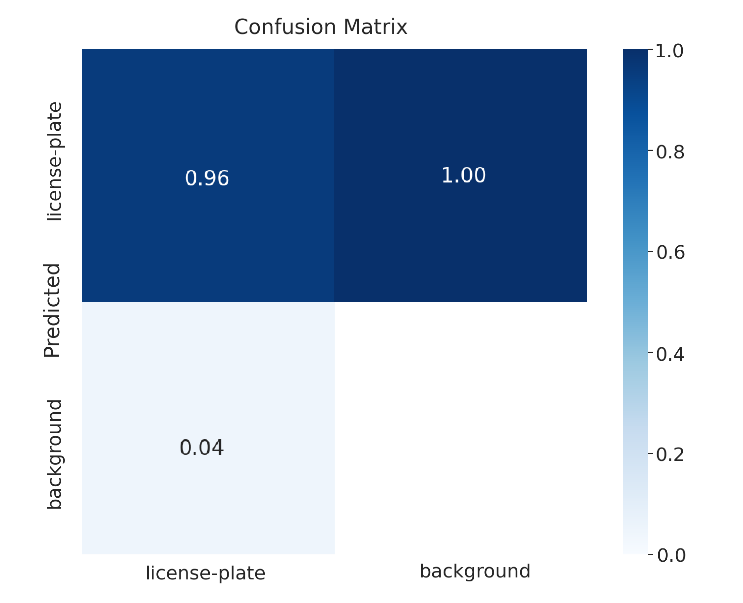


Fig. 7. Confusion Matrix between classes (license\_plate and background).

Fig. 8 and Fig. 9 show the Precision-Confidence and Recall-Confidence curves once the model’s been trained, while Fig. 10 and Fig. 11 show the Precision-Recall and F1-Confidence curves.

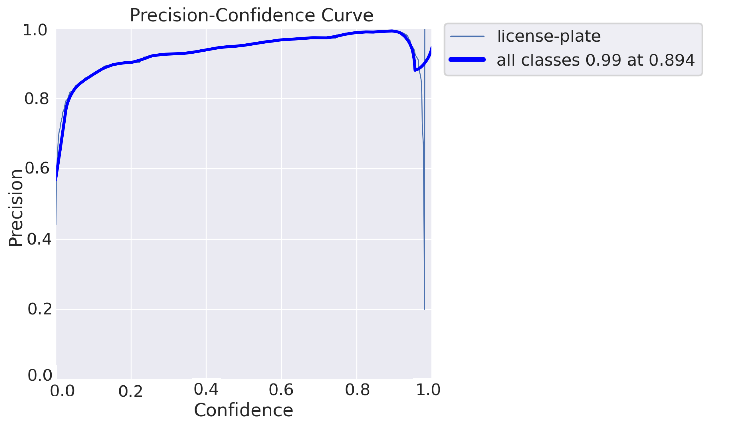


Fig. 8. Precision-Confidence curve for the license plate detection

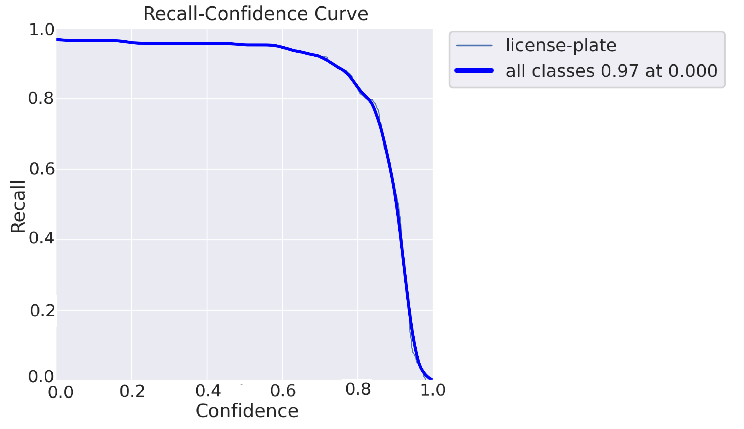


Fig. 9. Recall-Confidence curve for the license plate detection.

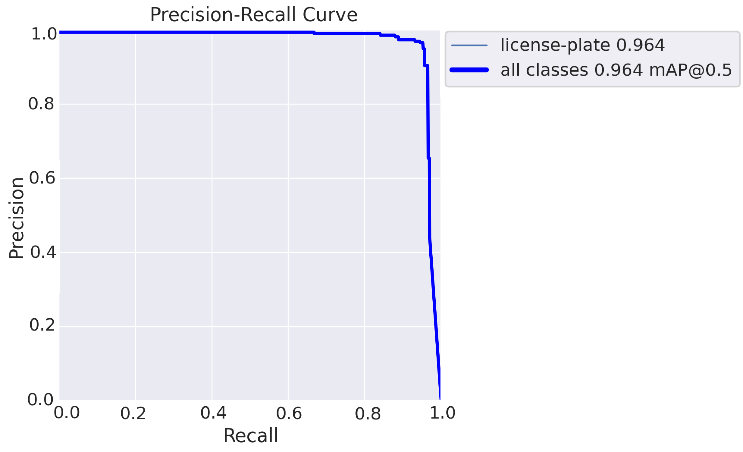
**

Fig. 10. Precision-Recall curve for the license plate detection

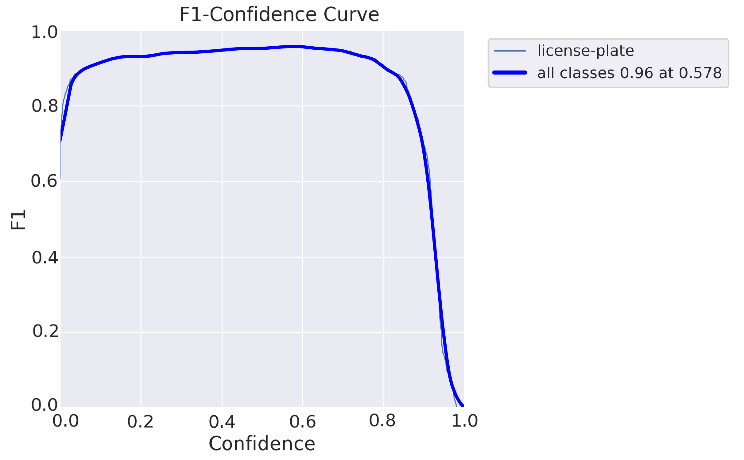


Fig. 11. F1-Confidence curve for the license plate detection.

A sample of images from the validation set has been fed to the model to obtain their prediction results, which have been displayed in Fig. 12 and Fig. 13. The model detects and recognizes the license plate among the background elements and outputs the probability of how certain it is in its prediction.

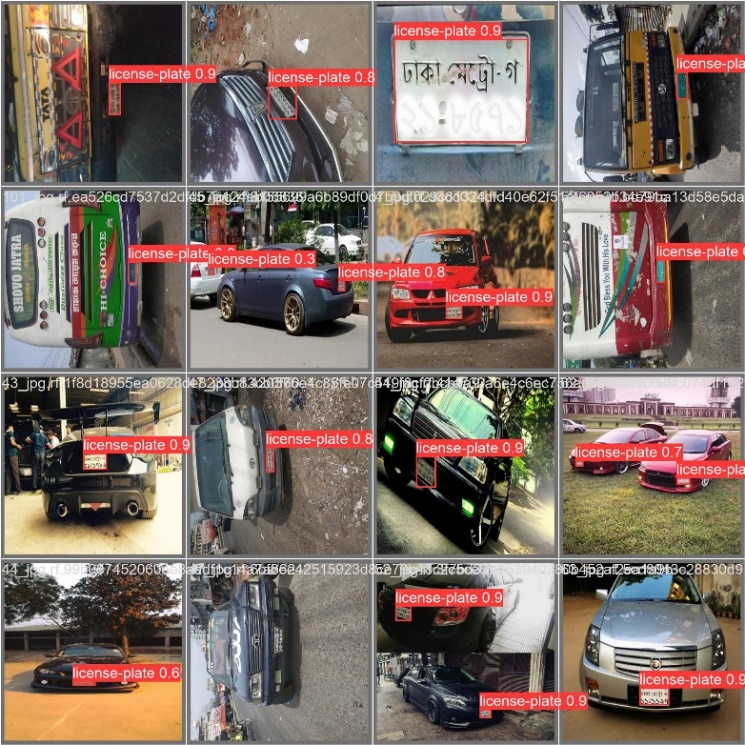
****

Fig. 1*2*: Model predictions on validation images (1).

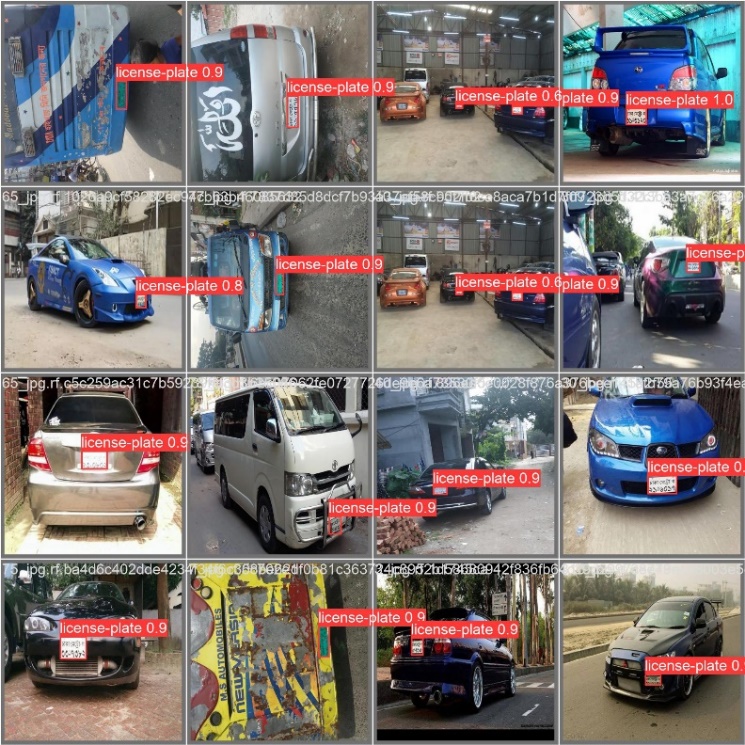
****

Fig. 1*3*. Model predictions on validation images (2).

## Results of EasyOCR

In the case of evaluating OCR results, accuracy is measured by comparing the output of the EasyOCR run of an image to its original version. The predictions of EasyOCR on a set of detected license plates from our test dataset have been displayed in Fig. 14 to Fig. 16. The coordinates of the plate are followed by the text reading and a confidence value for that reading for each line from the license plate.

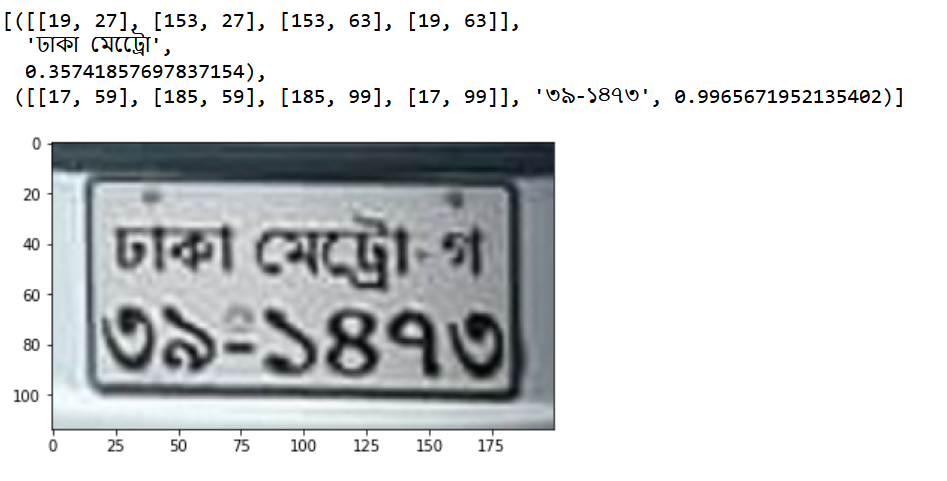
****

Fig. 14. EasyOCR results for “ঢাকা মেট্রো-গ” “৩৯-১৪৭৩”

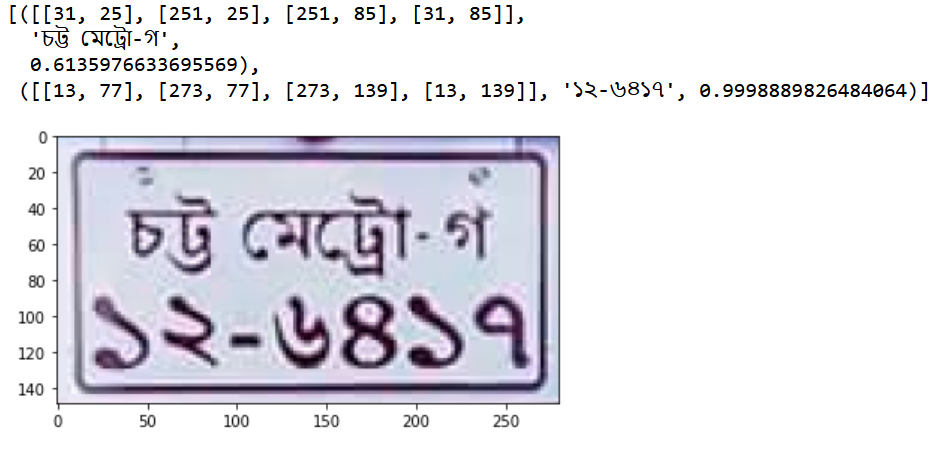
****

Fig. 15. EasyOCR Results for “চট্ট মেট্রো-গ” “১২-৬৪১৭”

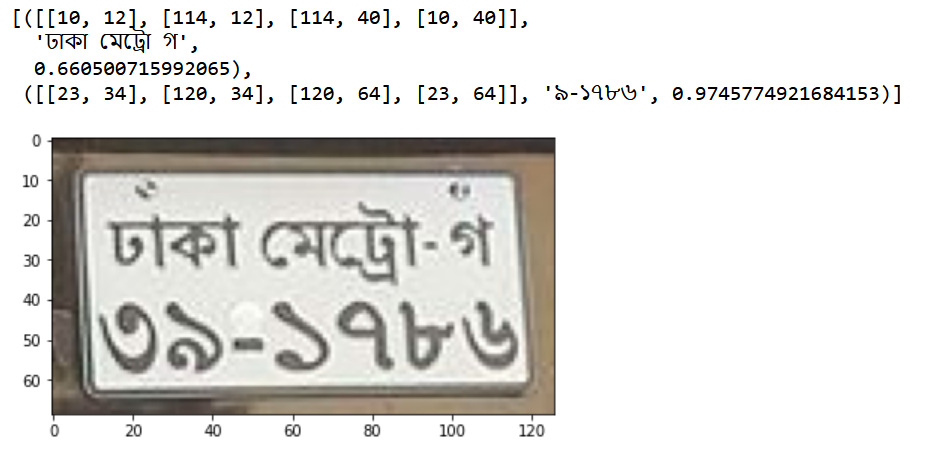
****

Fig. 16. EasyOCR results for “ঢাকা মেট্রো-গ” “৩৯-১৭৮৬”

EasyOCR correctly recognized the Bengali characters on most of the plates in the sample images, with a few cases of failure. Failures were more frequently seen in blurry, partially occluded, or skewed images. The failed instances can be removed by capturing images of better quality, resizing the images, removing the noise, and de-skewing the source image.

The performance of our proposed approach in license detection has been compared to those surveyed in the related works section in Table II. X represents values that were not found in the respective papers.

Table II: Comparison table of model performance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author** | **Dataset** | **Network** | **Accuracy** | **Recall** | **Precision** | **IoU** |
| [2] | Custom of 2,000 images | YOLOv3, computer vision, CNN, & Tesseract OCR | 88.89%. | 88.44% (Average) | 86.50% (Average) | >85% |
| [3] | UFPR-ALPR & SSIG | Fast-YOLO, YOLOv2 & CR-NET | 78.33% | 98.33% | 99% | 75% |
| [4] | Custom of 10,551 images | YOLOv2, YOLOv3, & CRNN-12 | 98.86% | 99.65% & 99.84% | 99.56% & 99.31% | 81.8% & 84.1% |
| [5] | Custom of 1,500 images and ISI Bangla Scene Character Database | YOLOv3 & ResNet-based CNN | 92.7% | X | X | >85% |
| [6] | Custom of 238 images | Sobel operator & Neural network | 91.47% | X | X | X |
| [7] | Custom of 1,000 images and 79 video clips | Haar-cascade classifier, MobileNet SSDv2, & Vision API | 82.7% | 59.3% | 63.6% | 50% |
| [8] | Custom of 1,000 | CNN | 74.7% | X | X | X |
| [18] | BRTA | CNN | 98.2% | X | X | X |
| This work | [9-13] and custom of 1,096 images | YOLOv8 & EasyOCR | 96.4% | 95.3% | 96.5% | >50% |

# 

# **Chapter 5**

# **Conclusion and Future Work**



## Conclusion

In this paper, a system for Bangladeshi vehicle number plate detection and recognition is developed. YOLOv8 was used to detect plates from still image inputs, and the plates were then extracted from those images. Finally, the characters are recognized using an EasyOCR reader. Our study made use of a dataset that consisted of images with various lighting conditions, multiple angles, and varied distances from the camera to the vehicles. Due to YOLOv8's advanced capabilities, it has performed extremely well in detecting plates and recognizing characters in such images.

In the future, we intend to add more images to our present dataset to train the model more extensively. We also plan to perform character segmentation using more advanced models such as CNNs or RCNNs.

# 

# **Bibliography**

References

1. Bangladesh Road Transport Authority. 2022. Motor Vehicle Registered: Bangladesh: Total. Accessed 20th January 2023. <https://www.ceicdata.com/en/bangladesh/motor-vehicle-registered/motor-vehicle-registered-bangladesh-total>
2. M. N. I. Suvon, R. Khan and M. Ferdous, “Real Time Bangla Number Plate Recognition using Computer Vision and Convolutional Neural Network”, International Conference on Artificial Intelligence in Engineering and Technology, 2020.
3. R. Laroca et al., “A robust real-time automatic license plate recognition based on the YOLO detector,” International Joint Conference On Neural Networks, pp. 1-10, 2018
4. H. Sun, M. Fu, A. Abdussalam, Z. Huang, S. Sun, and W. Wang, “License plate detection and recognition based on the YOLO detector and CRNN-12,” International Conference on Signal and Information Processing, Networking and Computers, pp. 66-74, 2019.
5. S. Abdullah, M. M. Hasan, and S. M. S. Islam, "YOLO-Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City," International Conference on Bangla Speech and Language Processing, 2018.
6. Miah, M. B. A., Akter, S., & Bonik, C. “Automatic Bangladeshi vehicle number plate recognition system using neural network”. Am. Int. J. Res. Sci. Technol. Eng. Math, 62-66, 2015
7. A. Ashrafee, A. M. Khan, M. S. Irbaz, and M. A. A. Nasim, "Real-time Bangla License Plate Recognition System for Low Resource Video-based Applications," Winter Conference on Applications of Computer Vision Workshops, 2022, pp. 479-488.
8. P. Marzuki et al., “A design of license plate recognition system using convolutional neural network,” International Journal of Electrical and Computer Engineering, vol. 9, 2019.
9. M. H. Tusar, M. T. Bhuiya, M. S. Hossain, A. Tabassum, and R. Khan, "Real Time Bangla License Plate Recognition with Deep Learning Techniques," International Conference on Artificial Intelligence in Engineering and Technology, 2022, pp. 1-6.
10. A. Yeafi (2022). Road Vehicle Images Dataset: Bangladeshi road vehicle images with YOLOv5 annotation. [Dataset] <https://www.kaggle.com/datasets/ashfakyeafi/road-vehicle-images-dataset>
11. N. Takbir, S. Afnan, and T. Ahmed (2020). Vehicle Detection - Bangladeshi Roads: Pictures with detection and classification bounding boxes for vehicles. [Dataset] <https://www.kaggle.com/datasets/nazmultakbir/vehicle-detection-bangladeshi-roads>
12. Nahiyan. Bangladeshi Cars with Number Plates Visible (2020). [Dataset] <https://www.kaggle.com/datasets/towniyan/bangladeshi-cars-with-number-plates-visible>
13. H. H. Shomee and A. Sams, “License Plate Detection and Recognition System for All Types of Bangladeshi Vehicles Using Multi-step Deep Learning Model,” Digital Image Computing: Techniques and Applications, 2021, pp. 1-7.
14. Jacob Solawetz. Jan 11, 2023. *What is YOLOv8? The Ultimate Guide*. Accessed 25th January 2023. <https://blog.roboflow.com/whats-new-in-yolov8/>
15. D. R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Saf, and D. Arun, “Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm,” International Conference on Electronics, Communication and Aerospace Technology, 2022, pp. 966-971.

1. *Easy OCR*. Accessed 5th February 2023.v <https://vitalvisiontechnology.com/easy-ocr/>
2. Aman Preet Gulati. 22nd June 2021. *Text detection from images using EasyOCR: Hands-on guide*. Accessed 25th February 2023. <https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/>
3. N.A.Alam, M. Ahsan, Md. A. Based, and J. Haider, “Intelligent System for Vehicles Number Plate Detection and Recognition Using Convolutional Neural Networks,” Technologies, vol. 9, no. 1, p. 9, Jan. 2021.
4. Shahin Akhter. *Unfit vehicles across Bangladesh on rise.* 24th March 2022*.* Accessed 23rd January 2023. <https://www.newagebd.net/article/166324/unfit-vehicles-across-bangladesh-on-rise>
5. Wikipedia. *Vehicle registration plates of Bangladesh.* Accessed January 2023. <https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Bangladesh>