Bengali License Plate Recognition System Using Deep Learning

Nawal Ayesha Khan   
1911301042North South Univeristy   
nawal.khan@northsouth.eduSarah Jasim  
1912260042North South Univeristysarah.jasim@northsouth.edu Seam E Nur Sami  
1912489042  
North South Univeristyseam.sami@northsouth.edu

*Abstract*— Automatic License Plate Recognition systems are a frequent topic of research all over the world, and play an important role in many applications, such as road traffic monitoring, law enforcement, automatic parking system, etc. Traffic violations and crime are prevalent in Bangladesh, so there's an urgent need for a suitable ALPR system.

In this paper we present a robust real-time system for Bangla license plate detection and recognition for license plates in Bangladesh. The system is robust as it has been trained and tested on a dataset containing images from multiple camera setups and environmental conditions. Our detection system utilizes the latest object detection algorithm YOLOv8, while combining it with EasyOCR. In our proposed architecture, there are two stages: first we have used YOLOv8 to detect or localize license plates. In the last stage, to recognize Bengali characters, we have utilized the EasyOCR package.

Results-

Conclusion-

# Introduction

Automatic License Plate Recognition (ALPR) or Vehicle License Plate Detection (VLPD) has been a frequent topic of research [3, 4, 5, 6, 7, 8] and plays an important role in many applications such as road traffic monitoring, law enforcement, automatic parking system, toll collection, access and border control, etc. It is the heart of intelligent transportation systems all around the world. ALPR systems are also of huge importance 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], leaving many more unregistered and unfit for operation. Furthermore, the number of unfit vehicles across the country has been on the rise over the years, often causing fatal road accidents. According to the Bangladesh Road Transport Authority, the number of vehicles without fitness certificates was 5.08 lakh as of January 2022 [18]. 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 vehicle registration plates which consist of various Bengali alphabets and numerals. According to its regulations, all license plates are written in Bangla language with a fixed two-line text format and colour for different types of vehicles The general format of Bangladeshi vehicle license plates "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, 6 numerals have been assigned to identify the vehicle, where the "12" field represents the vehicle class in Bengali numerals, while the "3456" field after the hyphen represents the unique vehicle number. The first line is in Bengali alphabets, while the second line is in Bengali numerals. The colours of the plates also hold some significance, for example- white is used for Private Service vehicles, while green license plates represent Public Service vehicles. License plates are generally installed in both the front and rear ends of the vehicle, the latter being a permanent attachment for the plate to the vehicle [2].

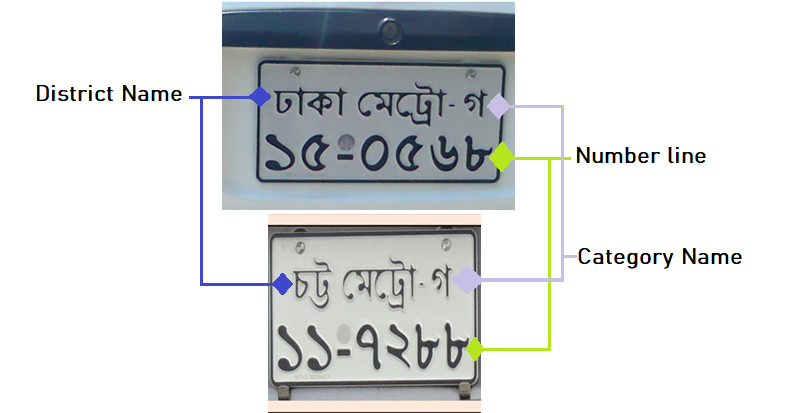
****

Figure : The standard BRTA license plate format. The first line of the license plate contains the name of the district and the word Metro in Bengali, followed by a hyphen and a character representing the vehicle class. The second line then contains six Bengali digits to identify the vehicle.

The letters permitted in Bengali vehicle registration plates are: 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

Whereas the numerals permitted in Bengali vehicle registration plates are: ০ ১ ২ ৩ ৪ ৫ ৬ ৭ ৮ ৯

In many cases, license plate images have been captured in ideal conditions, making them of not much use in bustling cities like Dhaka or Chittagong. Datasets containing Bangla license plates that represent non-ideal conditions, namely the presence of different lighting conditions, viewing angles, transparency, occlusion, etc., were unavailable. For this reason, it became necessary for us to compile a dataset representing these conditions and applying various levels of data augmentation to more accurately represent those non-ideal situations. Unlike the datasets for most other papers regarding this topic, we have also chosen to represent the plates of cities beside those of Dhaka city (such as Khulna, Chittagong, etc.) to allow for more variety in the data.

In this paper, we present a deep learning 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 different Bangladeshi vehicular license plate images that have been captured from various online datasets that resemble various real-world scenarios. In this regard, we have employed the latest YOLOv8 algorithm to successfully extract the license plate. To recognize the characters and numerals, we have also utilized the EasyOCR package. We have evaluated the performance of our model using precision, recall, standard loss curves, etc. Our proposed method achieves more than 85% Intersection over Union (IoU) in digit recognition. The YOLOv8 model achieves 92.7% accuracy in recognizing the Bangla character present in the license plate.

**In this work, an ALPR system has been developed using a DL approach- what u do, performance metrics- conclusion**

# Related Works

In this section, we briefly review several recent works that use DL approaches in the context of ALPR systems of various languages, including Bangla.

Many papers chose to use YOLO for the license detection portion and a CNN for the character recognition stage. In [3], a robust and efficient ALPR system based on the state-of-the-art YOLO object detector and CNNs were trained and fine-tuned for each ALPR stage to work under different conditions (e.g., variations in camera, lighting, and background). They also introduce a larger public dataset, called UFPR-ALPR dataset of Brazilian license plates, containing 150 videos and 4,500 frames captured when both the camera and various types of vehicles are moving. The UFPR-ALPR dataset is publicly available at <https://web.inf.ufpr.br/vri/databases/ufpr-alpr/>, subject to privacy restrictions. The resulting approach achieved impressive results in two datasets. First, in the SSIG dataset, composed of 2,000 frames from 101 vehicle videos, the system achieved a recognition rate of 93.53% and 47 Frames Per Second (FPS), a slight increase from both Sighthound and OpenALPR commercial systems (89.80% and 93.03%, respectively), and a significant improvement over previous results (81.80%). The results have been displayed in Figure X.

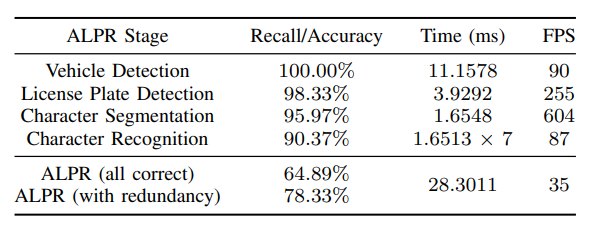


Figure : Results obtained and the computational time required in each stage in the UFPR-ALPR dataset.

Similarly, in [4], the paper focused on the detection and recognition of Chinese car license plate in complex background by comparing the performance of YOLOv2 and YOLOv3 on license plate detection. On the test data set, the recall and precision of both YOLOv2 and YOLOv3 exceeded 99% after 4,000 steps of training. The results showed that YOLOv2 performed well on detection speed, whereas YOLOv3 performed better on detection accuracy. In the recognition part, CRNN-12, a network that integrates the benefits of both CNN and RNN, is used to recognize license plate with different numbers of characters. The highest test accuracy achieved by the CRNN-12 was at 98.86% (9873/9987) at 60,000 steps, as demonstrated in Figure X below. They also used a custom dataset built from pictures taken in a variety of scenarios including highway crossings, road toll stations, city parking lot entrances, community entrances, etc. at different times of the day, multiple angles, levels of clarity, and various lighting situations.

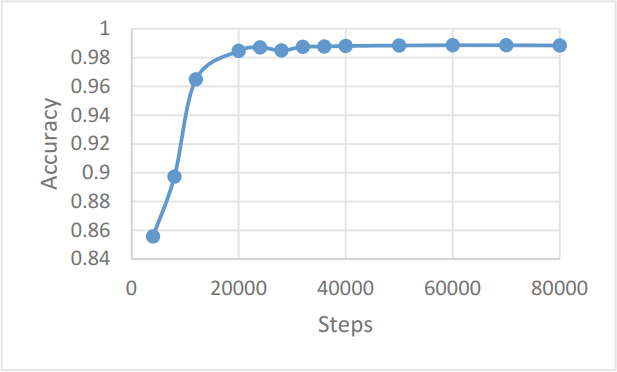


Figure : Test accuracy of CRNN-12 with different training steps

Another YOLO-based Bengali license plate detection network was designed in [5], where they employed YOLOv3 algorithm to successfully localize the license plate and recognize the digits. In most of the previous works related to Bangla license plate, images used for the ALPR were captured in ideal conditions due to unavailability of those representing non-ideal conditions. To overcome this, they built a Bangla scene character dataset containing more than 6,400 characters, with which they trained a ResNet-20-based deep CNN. The proposed method achieved more than 85% Intersection over Union (IoU) in digit recognition, while the ResNet-20-based CNN model achieved 92.7% accuracy in recognizing the Bangla characters present in the license plate. However, the license plate detection was based purely on a dataset of Bangla license plates of Dhaka metropolitan city, and was not tested on its performance on plates from the other districts. The output of their proposed network has been demonstrated in Figure X below. The proposed method correctly classified digits and characters with a high accuracy in different conditions, but still had a few failure instances, which have been marked with red.

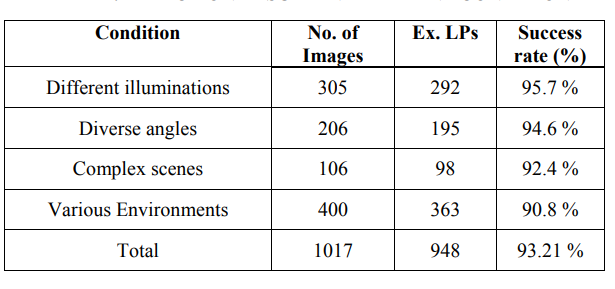


Figure : License plate recognition output of their proposed three-stage network. Red marks indicate characters that were identified incorrectly.

Other papers have expanded on various other methods for their ALPR systems. [6] designed a Smart Vehicle Number Plate Recognition System from photographs of Bangladeshi vehicle, where still photo of vehicles were used as input. Image enhancement is first performed using Contrast stretching, after which Sobel Operator is used for edge detection. After Character Segmentation, feature extraction is performed to obtain the unique features of every character, and neural network techniques are finally used for character recognition. Two groups of still images were collected for the dataset. However, the system faces many limitations, such as the performance decreasing for unclear, blur and very distant vehicle images. The system cannot detect and recognize the characters correctly in rainy and foggy weather, and can misidentify similar characters such as ‘0’ and ‘O’, ‘1’ and ‘I’, etc.

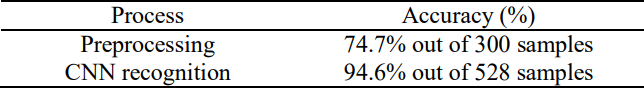
A similar paper, [7], designed a Bangladeshi vehicle digital license plate recognition system using support vector machine (SVM) for metropolitan cities (i.e. Dhaka, Chittagong). In the first phase, Sobel operator and histogram analysis are used to detect the license plate region, and connected component labeling and bounding box technique were then used to segment the characters of detected license plate region. Gabor filter has been used to extract the features, after which Kernel PCA reduces the high dimensionality of the obtained feature vector. Finally, SVM classifier is used to recognize the characters. More than 1000 images taken from various scenes were used, including complex scenes, diverse angles and different lightening conditions. The success rates achieved by their model in such conditions has been displayed in Table X below.

Table : Detection results in different conditions



The main contribution of [8] was to determine the best model for four-layered CNN architectures that have been used as the recognition method. This was accomplished by validating the best parameters of the enhanced Stochastic Diagonal Levenberg Marquardt (SDLM) learning algorithm and CNN network size. Several preprocessing algorithms such as Sobel operator edge detection, morphological operation and connected component analysis were used to localize the license plate, and then isolate and segment the characters before feeding the input to CNN. The images of vehicle license plate were randomly captured around the Malacca area in Malaysia to be used as the dataset. The license plate preprocessing stage achieved 74.7% accuracy and CNN recognition stage achieved 94.6% accuracy, as shown below in Table X. However, the methodology faced some limitations, as the preprocessing algorithm needs to be further improved in order to effectively filter out noises such as environment factor (illumination) to achieve higher accuracy. The preprocessing stage of this system needs improvement in order to achieve a higher accuracy level.

Table : Accuracy of their proposed LPR System



Through the process of this literature review, we have observed that in some countries, license plate characters are already in English, such as Brazilian license plates in [3], making them easier to recognize the characters. Working with license plates in other languages is difficult, such as in [4], but Bengali characters are even more challenging to recognize, as the language has many complexities such as strokes over the letters, similarities among some letters etc. Due to its high accuracy, we have determined the YOLO network to be ideal in detecting the license plate among still images of Bangladeshi vehicles.

# Methodology

For the license plate detection, we have used the newest version of YOLO (You Only Look Once) which is the YOLOv8, while the Bengali character recognition stage makes us of the EasyOCR package. We have also used the Roboflow tool to mark bounding boxes for the license plates dataset, as well as for the preprocessing stage, as has been detailed below.

## Dataset and Pre-processing

The dataset used in 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 [14] with visible Bangla license plates of almost all types of vehicles present in Bangladesh, excluding army, police, and government vehicles; as well as some collected from our own 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 all the datasets amounts to 4,369 files, many images were found to be unusable due to the extreme level of blurriness, overly pixelated images, or even the lack of a plate or vehicle in the image. Due to these issues, we had to include only relevant images for our research, leading to 1,096 images as the initial dataset. As YOLOv8 augments images during its training online, at each epoch, the model sees a slightly different variation of the images it has been provided. One of those augmentations is called mosaic augmentation, which involves stitching four images together so as to force the model to learn objects in new locations, in partial occlusion, and against different surrounding pixels. This resulted in our dataset ultimately containing 3,840 images in total.

## Deep Learning Models/Stages??

1. YOLOv8: The YOLO (You Only Look Once) series of models has become famous in the computer vision world due to its considerable accuracy while maintaining a small model size. YOLO models can be trained on a single GPU, making it accessible to a wider range of developers. Practitioners of Machine Learning and Deep Learning can deploy it for low cost on hardware or the cloud. [14]

YOLOv8 is the newest state-of-the-art YOLO model that can be used for object detection, image classification, and instance segmentation tasks. Developed by Ultralytics and launched on January 10th, 2023, it includes numerous architectural and developer experience changes and improvements over its well-known predecessor YOLOv5. It has a high rate of accuracy measured by COCO and Roboflow 100 and comes with a lot of developer-convenience features, from an easy-to-use CLI to a well-structured Python package [16]. The architecture used in YOLOv8 has been detailed below in Figure X.

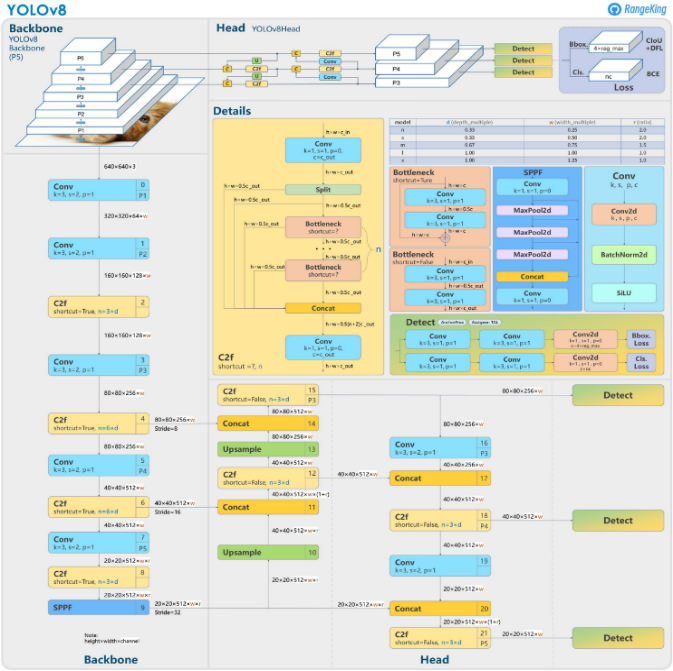


Figure 5: Architecture of YOLOv8 model

We originally intended to use the immediate predecessor of YOLOv8, the YOLOv7, for this project, but have since chosen to use the updated model to make use of its improved features.

The model was trained using Google Colab, which provides free access to powerful GPUs and requires no conﬁguration. After preparing the license plate dataset, we trained the model 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. We added the plate dataset and adjusted the number of epochs to be trained as well as the stack size to train the upper layers of the model to detect our classes. Training the YOLOv8 model for 300 epochs took about 2.52 hours. The hyper-parameters used for the YOLOv8 model have been detailed below in Table X.

Table : 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: Easy OCR is a font-dependent printed character reader based on a template matching algorithm. It has been designed to read any kind of short text (part numbers, serial numbers, expiry dates, manufacturing dates, lot codes, etc.) printed on labels or directly on parts. It is a python package that holds PyTorch as a backend handler. EasyOCR is created by the Jaided AI Company, and supports 42+ languages for detection purposes. [16].

EasyOCR requires training the font to be recognized. It is able to learn all possible characters to be read from sample images, making the recognition extremely flexible, fast and reliable. The training phase involves an interactive application used to show samples of the characters and allow the library to learn and store them in a font file [17].

# Results

## Metrics for Evaluation

The feasibility of the predictions made by our models will be cross-validated and critiqued through standard Precision, Recall, F1-scores, mAP, etc. The definition of these metrics are as follows:

1. Precision: It measures the proportion of positively predicted labels that are actually correct. The precision score is a useful measure of the **success of prediction when the classes are very imbalanced. It indicates how much we can rely on the model's positive predictions.**

***Precision Score =***

1. Recall: It represents the model’s ability to correctly predict the positives out of actual positives. A high recall score indicates that the model is good at identifying positive examples. Recall score can be used in the scenario where the labels are not equally divided among classes. It indicates any predictions that it should not have missed if the model is missing.

***Recall Score =***

The Precision-Recall curve is obtained by plotting the model's precision and recall values as a function of the model's confidence score threshold.

1. F1-Score: It is the harmonic mean of Precision and Recall; it therefore is commonly utilized as a classification evaluation metric due to weighing each metric evenly. This is a useful measure of the model in the scenarios where one tries to optimize either of precision or recall score and as a result, the model performance suffers.

***F1-Score =***

1. **mAP** (Mean Average Precision): It is a metric used to evaluate object detection models such as Fast R-CNN, YOLO, Mask R-CNN, etc. The mAP formula is based on the following sub metrics: Recall, Precision, Confusion Matrix, and Intersection over Union (IoU); the last of which indicates the overlap of the predicted bounding box coordinates to the ground truth box.

mAP= Pi

1. Distributional Focal Loss (dfl\_loss): DFL treats the continuous distribution of box locations as a discretized probability distribution. It is especially helpful in detection when the boundaries of the ground truth are blurred.
2. Classification Loss (cls\_loss): Classiﬁcation loss gives an idea of how well the algorithm can predict the correct class of a given object. This loss measures the correctness of the classification of each predicted bounding box. Each box may contain an object class, or a "background". This loss is usually called cross entropy loss.
3. Box Loss (box\_loss): The box loss represents how well the algorithm can locate the centre of an object and how well the predicted bounding box covers an object. It measures how "tight" or close the predicted bounding boxes are to the ground truth object.

## Results of YOLOv8 model simulations

The improvement in our model can be seen in the graphs from Figures X to Y, which display different performance metrics for both the training and validation sets.

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

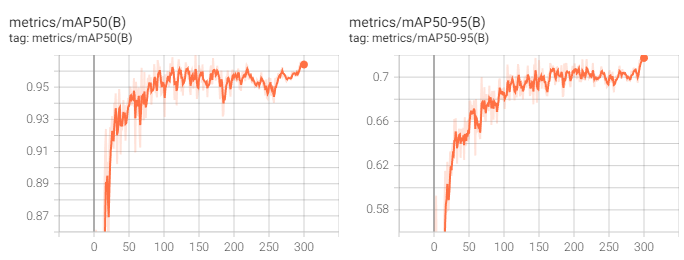


Figure : mAP50 curve (Left) and mAP50-95 curve (Right) for the license plate detection

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

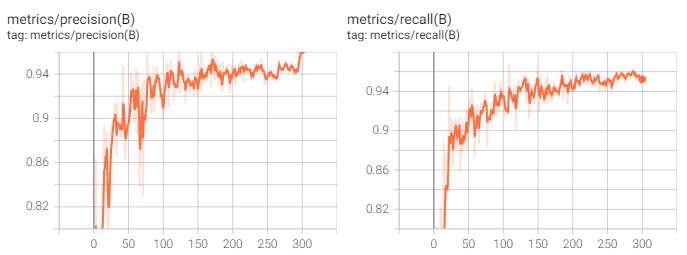


Figure : Precision curve (Left) and Recall curve (Right) for license plate detection

There are three different types of loss shown in Figure X: box loss, DFL loss and classiﬁcation loss. The curves show that after the 300 epochs during training, the box loss steadily drops to 0.540, the classification loss has a sharp decline to 0.285, and the DFL loss decreases to 0.285.

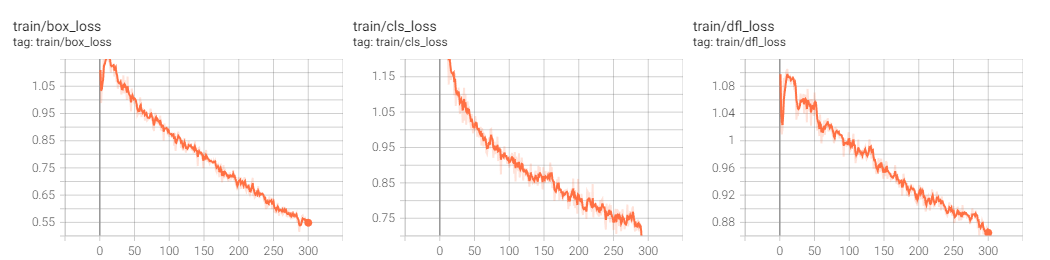


Figure : Box loss curve (Left), Classification loss curve (Middle), and Recall curve (Right) for license plate detection

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

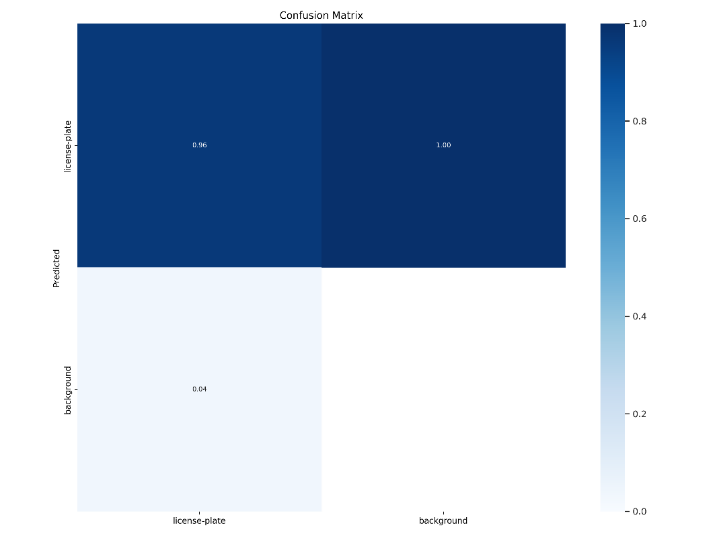
****

Figure : Confusion Matrix between classes (license\_plate and background)

Figure X shows the Precision-Confidence and Recall-Confidence curves during training, while Figure X shows the Precision-Recall and F1-Confidence curves.

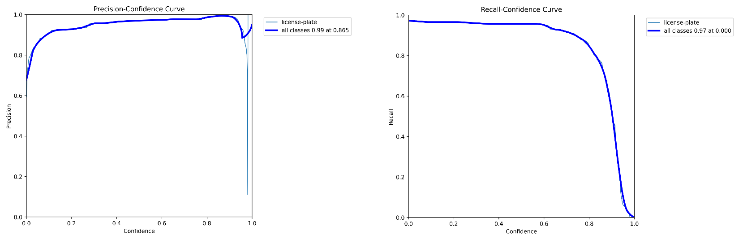
****

Figure : Precision-Confidence curve (Left) and Recall-Confidence curve (Right) for the license plate detection

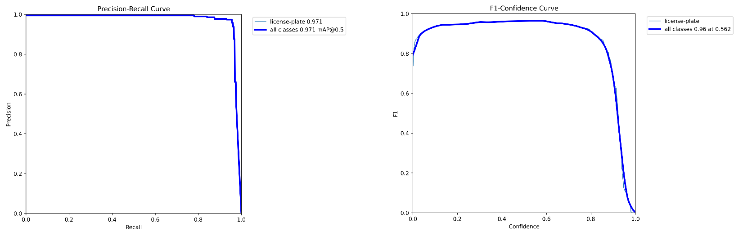
****

Figure : Precision-Recall curve (Left) and F1-Confidence curve (Right) for the license plate detection

The results of the model predictions on some of the images from the validation set have been displayed in Figures X and Y. The model detects and recognizes the license plate among the background elements, and outputs the probability of how certain it is in its prediction.

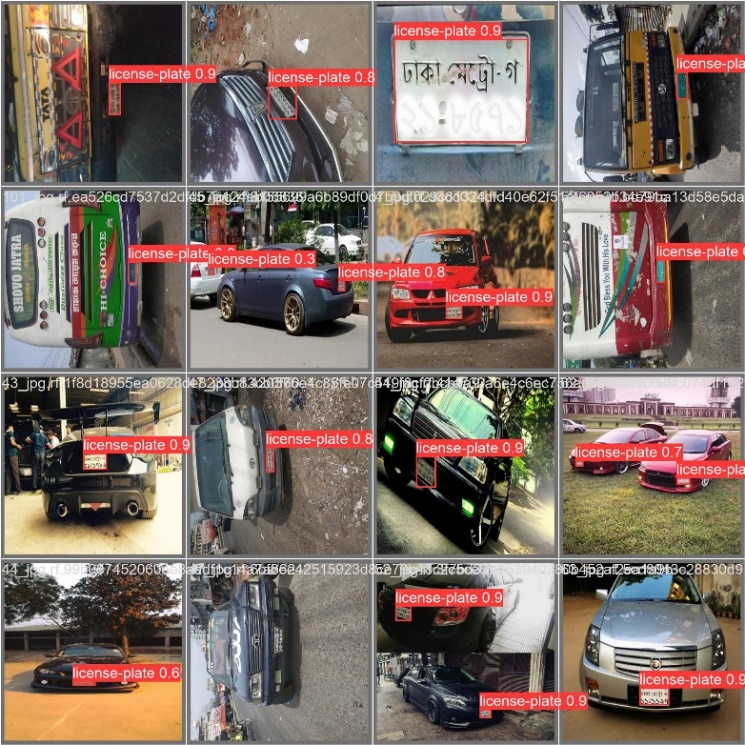
****

Figure : Model predictions on validation images (1)

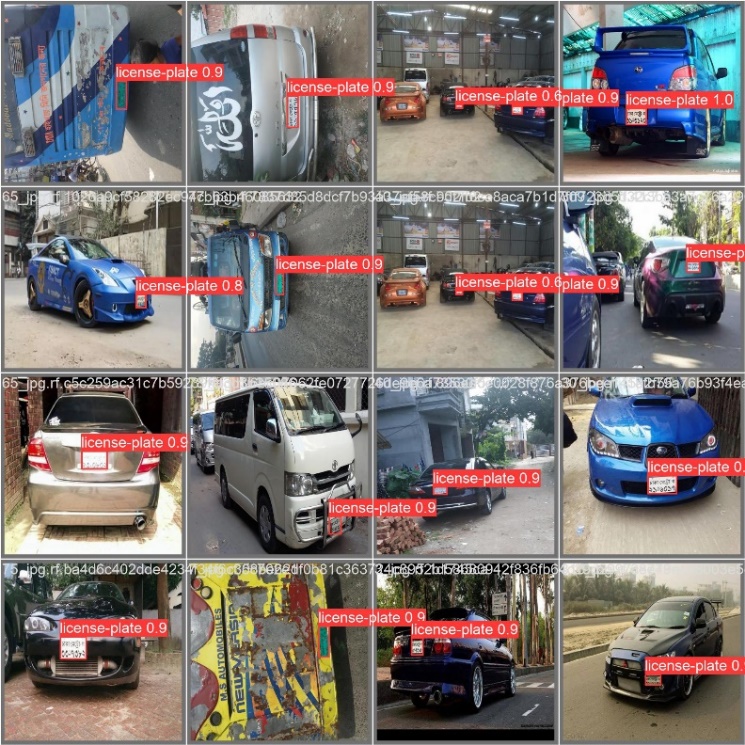
****

Figure : Model predictions on validation images (2)

## Results of EasyOCR

The predictions of the EasyOCR for a set of the detected license plates from the test images have been displayed from Figures X to Y. 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.

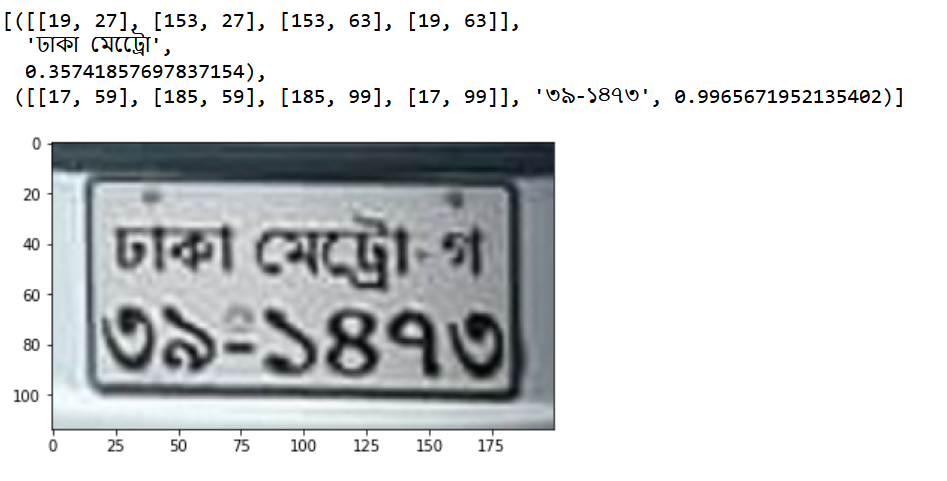
****

Figure : EasyOCR results for “ঢাকা মেট্রো-গ” “৩৯-১৪৭৩”

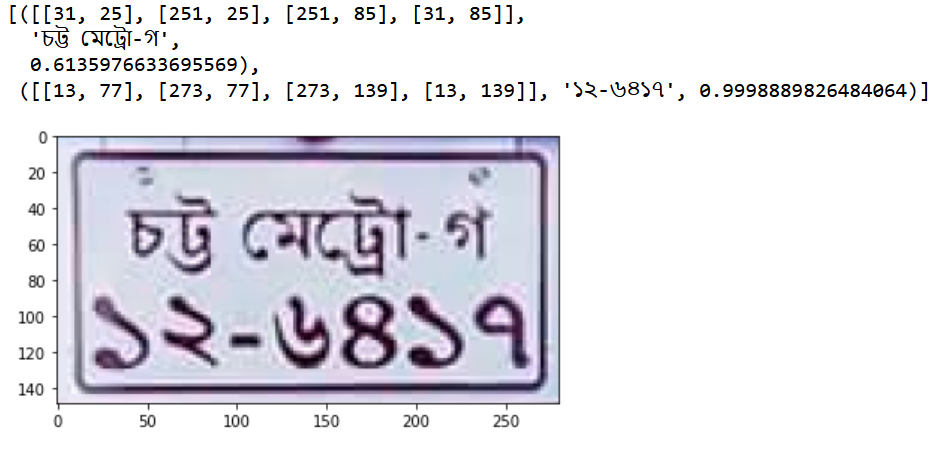
****

Figure : EasyOCR Results for “চট্ট মেট্রো-গ” “১২-৬৪১৭”

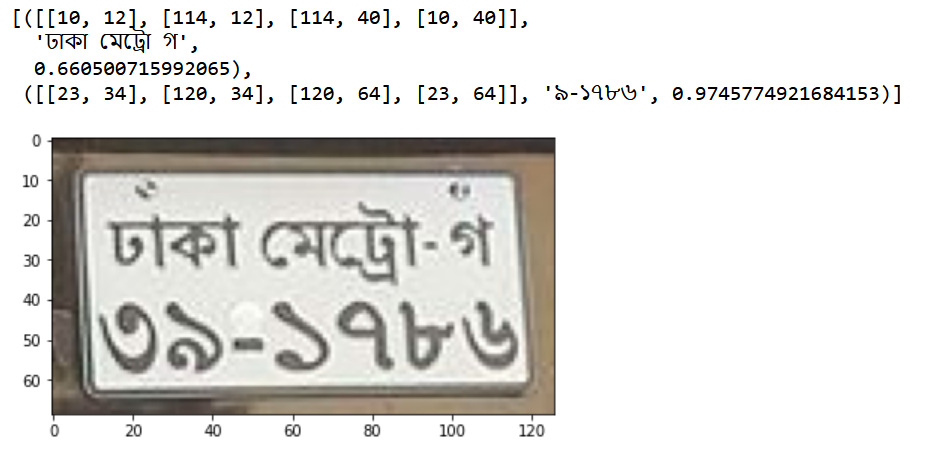
****

Figure : EasyOCR results for “ঢাকা মেট্রো-গ” “৩৯-১৭৮৬”

# Conclusion and Future Work

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

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. Wikipedia. *Vehicle registration plates of Bangladesh*. Accessed January 2nd, 2023 <https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Bangladesh>
3. Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 international joint conference on neural networks (ijcnn)* (pp. 1-10). IEEE.
4. Sun, H., Fu, M., Abdussalam, A., Huang, Z., Sun, S., & Wang, W. (2019). License plate detection and recognition based on the YOLO detector and CRNN-12. In *Signal and Information Processing, Networking and Computers: Proceedings of the 4th International Conference on Signal and Information Processing, Networking and Computers* *(ICSINC)* 4th (pp. 66-74). Springer Singapore.
5. Abdullah, S., Hasan, M. M., & Islam, S. M. S. (2018, September). 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.
6. Miah, M. B. A., Akter, S., & Bonik, C. (2015). Automatic Bangladeshi vehicle number plate recognition system using neural network. *Am. Int. J. Res. Sci. Technol. Eng. Math*, 62-66.
7. Uddin, M. A., Joolee, J. B., & Chowdhury, S. A. (2016). Bangladeshi vehicle digital license plate recognition for metropolitan cities using support vector machine. In *Proc. International Conference on Advanced Information and Communication Technology*.
8. Marzuki, P., Syafeeza, A. R., Wong, Y. C., Hamid, N. A., Alisa, A. N., & Ibrahim, M. M. (2019). A design of license plate recognition system using convolutional neural network. *International Journal of Electrical and Computer Engineering*, *9*(3), 2196.
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," *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2022, pp. 1-6.
10. Ashfak Yeafi (2022*). Road Vehicle Images Dataset: Bangladeshi road vehicle images with YOLO v5 annotation*. [Dataset] <https://www.kaggle.com/datasets/ashfakyeafi/road-vehicle-images-dataset>
11. Nazmul Takbir, Sihat Afnan, Tanjim 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. Ataher Sams, & Homaira Huda Shomee. (2021). Bangla LPDB - A (Version v1) [Data set]. *International Conference on Digital Image Computing: Techniques and Applications (IEEE DICTA)*, Gold Coast, Queensland Australia. Zenodo. <https://doi.org/10.5281/zenodo.4718238>
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. Shipra Saxena. March 19, 2021. *Introduction to The Architecture of Alexnet*. Accessed 3rd February, 2023. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/>
16. *Easy OCR*. Accessed 5th February, 2023.v <https://vitalvisiontechnology.com/easy-ocr/>
17. 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/>