Amplifying and Enhancing Safety with Real-Time License Plate Recognition for Nepal's Traffic System Using YOLOv8

Saishna Budhathoki
Department of Software Engineering
Nepal college of Information Technology
Kathmandu, Nepal
saishna.201729@ncit.edu.np

Er. Thomas Basyal
Department of Software Engineering
Nepal college of Information Technology
Kathmandu, Nepal
thomas.basyal007@gmail.com

***Abstract*—As the rate of urbanization in Nepal rises, there are problems with regulating and controlling the movement of vehicles and efficient policing. In this regard, we exhibit an enhanced License Plate Recognition (LPR) system which is appropriate for the Nepalese environment. The system also uses deep learning techniques, especially YOLOv8, for detecting Nepali Devanagari characters on the license plate. Our LPR system is designed to fulfill the requirements of Nepali vehicles, which include two-wheelers, four-wheelers, and license plates of various designs and scripts. Real-time image processing and OCR form the system that enables the precise and fast recognition of the marks. The measures for increasing road safety, traffic regulation, and supporting law enforcement agencies are based on the ideas of smart technologies. This paper describes the background of the LPR system, its work, and its potential application in enhancing security and order in Nepalese cities.**

Keywords— License Plate Recognition; Real-time image processing; OCR; Enhancing Security

I. Introduction

Automated License Plate Recognition (ALPR) systems are crucial in modern vehicle management and are applied for law enforcement and traffic regulation. Such systems have evolved over the years, particularly with the assistance of computer vision and machine learning technology. The objective of this research is to develop an LPR system for Nepali vehicles which is very much difficult due to the use of Devanagari script and different format of license plates in Nepalese. These are some of the challenges that must be appreciated to have an efficient and accurate recognition system

In the context of Nepal, it is rather peculiar for LPR systems because the Nepalese use the Devanagari script for their license plates, while most of other countries employ Latin script. Compared to Latin script, Devanagari is more complicated due to the difference in the set of characters and, therefore, requires specific methods for its recognition and analysis. Earlier works have revealed that the standard techniques of OCR, which are designed for Latin script, perform poorly in the case of Devanagari script because of its complexity and flexibility in license plate format [1]. Thus, the localized approach integrating the recent developments in deep learning and computer vision is crucial for the successful implementation of the LPR in Nepal.

The state-of-art deep learning especially the CNNs and YOLO architectures have been found to be very promising in object detection and recognition. These methodologies provide sound approaches to design LPR system for processing the intricacies of Devanagari script [2]. In this study, we use YOLOv8 model for real-time object detection as it is one of the most efficient models and modify it to detect the characters and pattern of Nepali license plates. This entails feeding the model with a dataset that has been labeled for Nepali license plates with consideration of different designs of the plates and the environmental conditions.

It is possible to increase the accuracy of LPR system by uniting OCR with deep learning models such as YOLOv8. Thus, the proposed solution combines the object detection of YOLOv8 with YOLOV8 OCR's character recognition to obtain the best results. Thus, based on the comparison of YOLOv8 with other models such as traditional CNNs and standalone OCR, we define the best practices for the given application. Based on our results, it is concluded that YOLOv8 is more precise and efficient than the other methods, which makes it ideal for real-time LPR in Nepal [3].

A. *Research Objective*

1) To overcome the difficulties of identifying and detecting alterations in Devanagari license plates.
2) To execute system tracking invalid license plate,speed monitoring and mechanize fine charges by associating vehicle data to the citizen's national ID

II. Literature Review

The paper by Dawadi, Bal, and Pokharel on "Devanagari License Plate Detection, Classification, and Recognition" directly targets the problems of detecting and recognizing the Devanagari script used on Nepalese license plates. For Latin characters, there are established methods, but for Devanagari plates, certain procedures have to be employed because of the complicated shapes and changes. For accurate detection, YOLO8 and CNNs are used while SORT is used for real-time tracking. Thus, through the analysis of the structure and characteristics of these plates, the study provides relevant findings and recommendations that can be used in practice to improve the recognition rate in this case [4].Gnanaprakash et al. study fills the gap of vehicle tracking as the number of vehicles increases and the need for real-time monitoring arises. They suggest an automatic system based on YOLO for object recognition that transforms the video stream into images, detects automobiles, recognizes plates, and recognizes characters. Utilizing the ImageAI library, their model achieved impressive accuracy rates: Car detection was 97 percent accurate, license plate localization was 98 percent accurate, and character recognition was 90 percent accurate. Such an approach seems to be a rather effective solution to enhance security and traffic conditions in urban environments [5]. The paper offers the solution to number plate identification challenges in the challenging environment using Faster R-CNN, the enhanced object detection model. The system enhances the detection efficiency through frame segmentation and image interpolation It also employs OCR for plate recognition. With an impressive 99. It has 1% of accuracy and is helpful in such issues as distortion of the perspective and differences in lighting. The findings of this research offer a clear solution to the improvement of traffic monitoring and policing and its relevance in the urban and transport areas [6].The ALPR is presented with a promising improvement in Masood et al.'s "License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks". As for the issues of the variations, the study employs CNNs that help to avoid the problems of illumination and occlusion that were critical in the previous studies. The features are learnt directly from raw images hence no complicated preprocessing as in the proposed CNN model. The findings of the study reveal how CNNs can improve the license plate recognition systems and therefore approves the idea of further improvement in the future [7].

III. Research Gap

There are many researches conducted on LPR systems around the world, the problem of Nepali license plates with Devanagari script, regional variation, and varying real-world conditions has not been addressed. This research aims to develop an LPR system for these characteristics and present the comparison of different OCR techniques and the actual use of the LPR system in Nepal.

IV. YOLOV8 Architecture

A. Introduction

YOLOv8 (You Only Look Once, Version 8) is an advanced object detection model. It divides an input image into a grid and predicts bounding boxes, confidence scores, and class probabilities for each grid cell. This document provides the mathematical implementation of the YOLOv8 architecture.

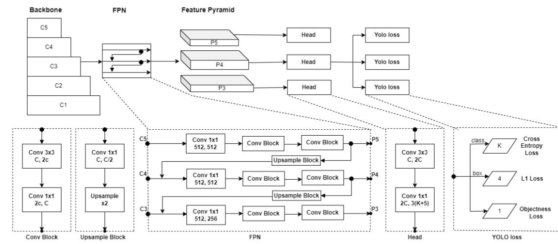

Fig. 1. Yolov8 Architecture

B. Input Image and Grid

The input image is divided into an $S \times S$ grid. Each grid cell is responsible for predicting objects whose centers fall within the cell.

C. Bounding Box Prediction

Each grid cell predicts B bounding boxes. For each bounding box, the following parameters are predicted:

- **Center Coordinates:** (x, y)
$$x = \sigma(t_x) + c_x$$
$$y = \sigma(t_y) + c_y$$
where σ is the sigmoid function, and (c_x, c_y) are the top-left corner coordinates of the grid cell.
- **Width and Height:** (w, h)
$$w = p_w \cdot e^{t_w}$$
$$h = p_h \cdot e^{t_h}$$
where p_w and p_h are the anchor box dimensions.
- **Objectness Score:** p_c
$$p_c = \sigma(t_c)$$

D. Class Prediction

For each bounding box, the model predicts a set of class probabilities $p_1, p_2, \ldots, p_C$:

$$p_i = \frac{e^{s_i}}{\sum_{j=1}^{C} e^{s_j}}$$

where s_i is the score for class i.

E. Loss Function

The loss function consists of three components:

Localization Loss (Bounding Box Regression Loss)

$$\text{Loss}_{\text{loc}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \Big[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \Big]$$

F. *Confidence Loss (Objectness Loss)*

$$\text{Loss}_{\text{conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} (p_{ij} - \hat{p}_{ij})^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{noobj}} (p_{ij} - \hat{p}_{ij})^2$$

G. *Class Probability Loss*

$$\text{Loss}_{\text{class}} = \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

V. System Implementation

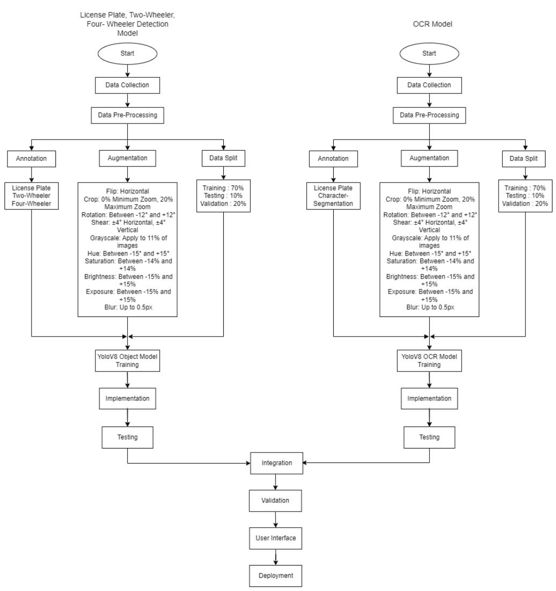

Fig. 2. Flowchart of System Implemetation

A. *Implementation of YOLOV8 model for License Plate Detection*

In the case of License Plate Detection System, the data collection from kaggle [4] was carried out to involve a set of pictures that was inclusive of about 16000 pictures with varying light intensity, angles, and backgrounds. The acquired data had both distinguishable and indistinct license plates, two-wheelers, and four-wheelers and other types of vehicles. The dataset was labeled using LabelImg tool to label the regions of interest such as license plates, two-wheelers, and four-wheelers. Horizontal flipping, cropping, rotation and changes in hue, saturation, brightness were used to expand the dataset before it was used to train the YOLOv8 model.

From this dataset the training sets have been taken as 70%, the validation set as 20% and the testing set as 10%. Hence, based on metrics like precision recall mean average precision (mAP), YOLOv8 model for detecting License Plates together with Two-Wheelers and Four-Wheelers was evaluated. The results demonstrated a high level of accuracy in the identification of License Plates as well as Four Wheelers that are stored within various confidence intervals. But, the Two Wheelers detection was quite volatile as compared to the above mentioned companies. A high mAP@0.5 of 0.902 also showed that the model has the ability to generalize well and overall, perform well.

1) *Confusion Matrix*

TABLE I
CONFUSION MATRIX

Class	TP	FP	FN
Four_Wheeler	12	1 (LP)	2 (BG)
License_Plate	22	2 (BG)	1 (TW)
Two_Wheeler	8	1 (BG)	1 (LP)
Background	-	-	-

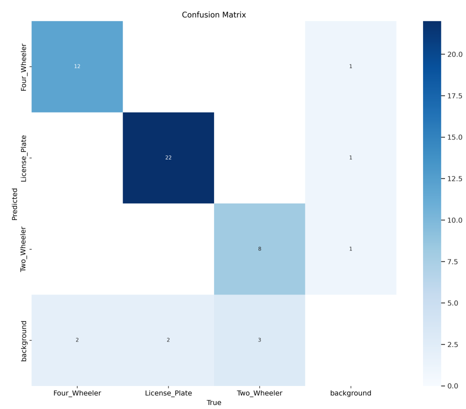

Fig. 3. Confusion Matrix for Licenseplate, Two Wheeler, Four-Wheeler Model

The confusion matrix shows the result of the model's prediction for each class. For Four_Wheeler it correctly recognized 12 but failed to recognize 1 as License Plate and 2 as background. It was noted that License_Plate was identified with high accuracy 22 times, 2 of which were confused with the background class, and 1 with the Two_Wheeler class. Two_Wheeler had a total of 8 detections, out of which 1 was misclassified as background and 1 as License Plate. The Background class was also heavily misclassified which suggest that the model struggles to distinguish between actual classes and the background.

2) *Recall Confidence Curve*

Recall-Confidence graph shows the level of recall of the model in relation to the confidence level for different classes. For Four-Wheelers, the recall is still high for most of the levels of confidence but drops sharply at higher levels of confidence. License Plates in general have a very high recall, almost 1.0, however, it decreases as the confidence level gets higher, which shows that the model is sensitive to the confidence thresholds. In the same regard, Two-Wheelers record a relatively low recall compared to the other classes, with the recall decreasing at a slower rate as the confidence levels increase. This implies

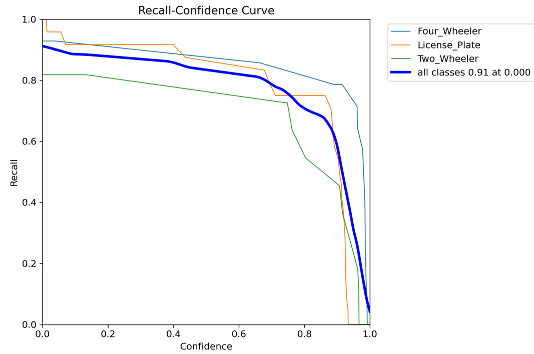

Fig. 4. R-curve for object Model

that although the model is very effective in recognizing Four-Wheelers and License Plates, it is relatively less effective in recognizing Two-Wheelers particularly when the confidence level is high.

3) F1 Curve

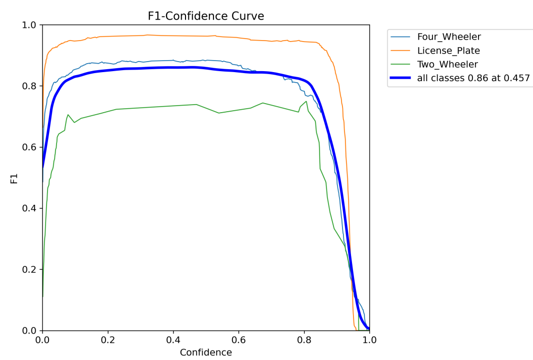

Fig. 5. F1-curve for object Model

The F1-Confidence Curve illustrates the relationship between confidence levels and the F1 scores for three classes: They are Four_Wheeler, License_Plate, and Two_Wheeler. Among all the classes, License_Plate has the best F1 scores at different confidence levels, and Two_Wheeler has the lowest F1 scores and more fluctuations. The thick blue line shows the average F1 score which is an overall performance and this is the reason why the presented model achieves the best result with the value of 0.86 with the specified confidence level of 0.457.

4) Precision Recall Curve

PR curve defines the relation between precision and recall; Four-Wheelers obtained high precision of 0.924. License Plates have the amongst the highest accuracy levels (0.968 Mean) with almost negligible accuracy shrinkage as the ramp for recall goes up, thus proving the resilience of the model. Nonetheless, the Two-Wheelers category has lower precision (0. 815) and higher decline of the precise with the increase of the recall which indicates a higher percentage of false positive elements; yet, the overall model has high effectiveness with the mean Average Precision (mAP) equal to 0. 902, which

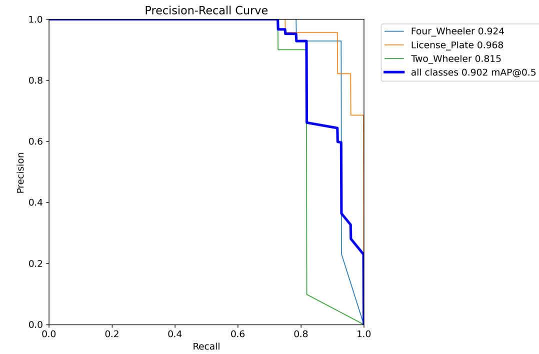

Fig. 6. PR curve for object Model

indicates that for all the classes of the inventory it has the right proportion.

5) Precision Confidence Curve

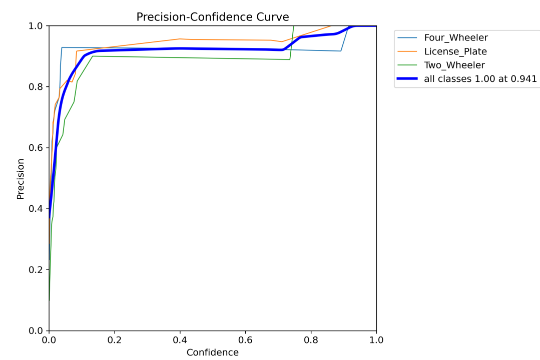

Fig. 7. P curve for object Model

The precision for FourWheelers begins with a lower value but quickly rises and levels off at approximately 0.9 which is a good score in all the confidence intervals. License_Plate seems to have high precision at all confidence levels and Two_Wheeler has comparatively less precision at the start which increases with some fluctuation indicating its lesser stable nature at low confidence levels.

B. Implementation of YOLOV8 OCR

1) Data Preprocessing

In case of License Plate Recognition (LPR) system, we collected a set of Nepali license plate characters from Kaggle site [8] and cropped some more characters by our selves. The collected data was then annotated using the jTessBoxEditor-2.To enhance the stability of the dataset, the following augmentations were applied: horizontal flips, random crop, rotations, shear transformations, grayscale conversion, change to hue, saturation, brightness, exposure, random noise, and blurring. These augmentations enhanced the training data set's diversity, which was beneficial in addressing problems of font, size, color, and background noise in the OCR engine.

2) Model Comparison

We evaluated two models: The algorithms that have been used are Tesseract combined CNN, and YOLOv8 due to their ability to identify characters in Nepali license plates. The results obtained for the models were precision, F1 score, average precision (AP), and global accuracy. Out of two models, YOLOv8 was the most accurate with a precision of 1.00, F1 score of 0. 89, and AP of 0. 906 (mAP@0 .5) scoring the accuracy of approximately 91%. Tesseract's accuracy was slightly lower with an accuracy of 67%. Hence, Yolov8 model have higher Recall and Precision making model efficient for OCR and Character Recognition

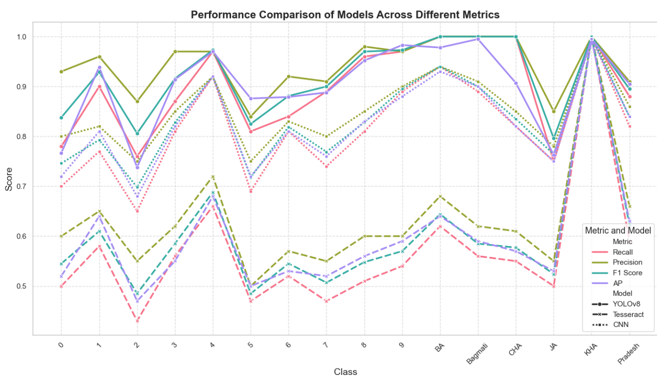

Fig. 8. Curve Comparsion between yolov8 and tesseract with CNN Model

Fig. 9. Comparision between yolov8 and tesseract output

3) Model Training

The entire dataset was split into training, validation and test data set with the ratio of 70:20:10 and the YOLOv8 model was trained for 100 epochs with batch size of 16. It was able to identify 22 classes of Devanagari Character, digits, and special labels including 'BA', 'Bagmati', 'CHA'. During the evaluation of the model, the YOLOv8 model received high recall rate, precision rate and F1 score, moreover, the confusion matrix depicted high accuracy of most of the classes. The F1-confidence, precision-confidence, precision-recall, and recall-confidence plots also depicted how YOLOv8 was a good balance between precision and recall and quite stable for practical application for NLPD.

4) YOLOv8 Model Output

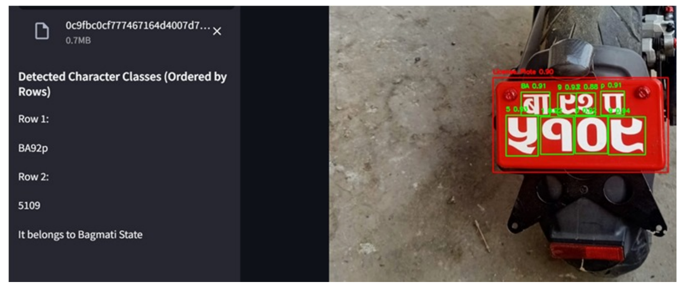

Fig. 10. output of the YOLOv8 Model

5) Confusion Matrix

TABLE II
CONFUSION MATRIX

Class	TP	FP	FN
0	39	11 (background)	1 (3)
1	47	5 (background)	1
2	35	8 (background)	1
3	28	5 (background)	3
4	37	3 (background)	1
5	26	3 (background)	1
6	26	5 (background)	1
7	42	3 (background)	
8	45	3 (background)	
9	36	2 (background)	3
BA	48	2 (background)	
Bagmati	9	16 (background)	
CHA	22	1 (background)	1
JA	3		
KA			1 (background)
KHA			
Pradesh	15	2 (background)	
jha		1 (background)	
p	32	3 (background)	
pra			
sa			
ya			
background	-	-	-

The confusion matrix shows the TP, FP, and FN of each class of the classifier: For instance, misclassification labels such as the ones in the FP column for the 'background' class are recorded. This matrix helps in understanding the precision of the model and areas that needs more precision.

6) Precision-Recall Curve

Precision-Recall (PR) curve for each class and an overall mean Average Precision (mAP) of 0. 906 and high recall at low confidence, which proves high accuracy but at the same

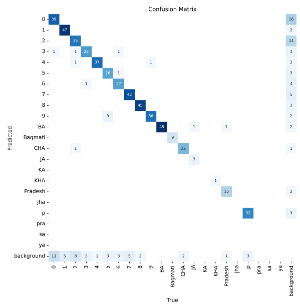

Fig. 11. Confusion Matrix for OCR

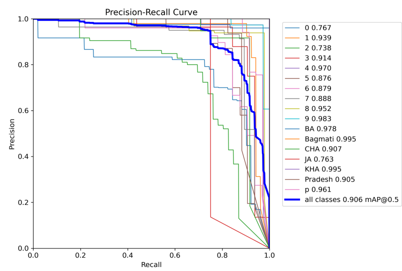

Fig. 12. PR Curve for OCR

time points out that some classes such as '2' and 'JA' needs enhancement in training.

7) Recall Confidence Curve

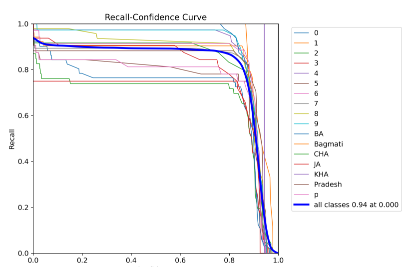

Fig. 13. R Curve for OCR

The Recall-Confidence Curve shows the level of recall and its dependency on the confidence level and reveals that while the recall is high for low confidence level, it rapidly decreases with the increase in the threshold level and is different for different classes and overall score of this curve is 0.94 at 0.000

8) F1 Curve

The F1-Confidence curve reveals that the F1 score, that is the harmonic mean of precision and recall, attains the highest value of the optimal confidence threshold of 0.754 and then decreases, However, the F1 score average in all is 0.89.

VI. Conclusion

The objective of the LPR system was to detect license plates, two-wheelers and four-wheelers and recognize devangari characters from License Plate.The system accuracy was 93% when using YOLOv8 model, which has a slightly

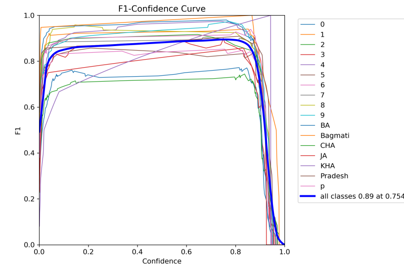

Fig. 14. F1 Curve for OCR

higher accuracy than using Tesseract and CNN that had only 60% accuracy in OCR and recognition. These disparities showed a need for a better solution, this led to the integration of YOLOv8 for OCR. They have been both adopted and incorporated in Streamlit to deliver the recognition aspects for such intended applications dependably and efficiently. Thus, The project establishes that it is necessary to employ elaborate datasets and complicated machine learning models in increasing superior performance of automatic license plate recognition systems.In the forthcoming work, we will be implementing and carring out License Plate Recognition(LPR) for speed spotting, and forgery license plate number identification and safeguarding.

Acknowledgment

The authors would like to extend sincere thanks to Nepal College OF Information Technology, Balkumari Lalitpur for providing support during model training, and implementation with required technology. The author would also like to thank the Department of Software Engineering for their encouragement during the system development and study. This research work has been carried out in the Department of Software Engineering, Nepal College Of Information Technology

References

[1] Adapting the Tesseract open source OCR engine for multilingual OCR — Proceedings of the International Workshop on Multilingual OCR — dl.acm.org, https://dl.acm.org/doi/10.1145/1577802. 1577804, [Accessed 01-08-2024]. 100

[2] edmon , J . andFarhadi , A . (2018) YOLOv3 : AnIncrementalImprovement . ComputerScience , arXiv : 1804 . 02767 . http : / / arxiv . org / abs / 1804 . 02767, [Accessed 01-08-2024]

[3] https : / / www . researchgate . net / publication / 220861931 _ Comparative _ Study _ of _ Devnagari _ Handwritten _ Character _ Recognition _ Using _ Different _ Feature _ and _ Classifiers, [Accessed 01-08- 2024.

[4] P. R. Dawadi, B. K. Bal, and M. Pokharel, "Devanagari license plate detection, classification and recognition," in First International Conference on Advances in Computer Vision and Artificial Intelligence Technologies (ACVAIT 2022), Atlantis Press, 2023, pp. 304–318.

[5] Automatic number plate recognition using deep learning - IOPscience — iopscience.iop.org, https://iopscience.iop.org/article/ 10.1088/1757-899X/1084/1/012027/meta, [Accessed 04-05- 2024].

[6] N. Ap, T. Vigneshwaran, M. Arappradhan, and R. Madhanraj, "Automatic number plate detection in vehicles using faster r-cnn," in 2020 Interna tional Conference on System, Computation, Automation and Networking (ICSCAN), 2020, pp. 1–6. DOI: 10.1109/ICSCAN49426.2020. 9262400.

[7] S.Z.Masood,G.Shu,A.Dehghan,andE.G.Ortiz,“Licenseplate detection and recognition using deeply learned convolutional neural networks,” arXiv preprint arXiv:1703.07330, 2017.
[8] Vehicle Number Plate Dataset(Nepal) — kaggle.com, https://www.kaggle.com/datasets/ishworsubedii/vehicle-number- plate-datasetnepal/data, [Accessed 04-05-2024.