End-To-End YOLOv12-Based Multi-Stage Pipeline for Bangla License Plate Recognition

Iffat Ara Nowshin

Bachelor of Science

DAFFODIL INTERNATIONAL UNIVERSITY

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Iffat Ara Nowshin
Date of Birth	:	04-04-2002
Title	:	YOLOv12-Based Dual-Model Architecture for End-to-End Bangla License Plate Recognition
Academic Session	:	2021-2025

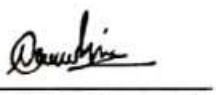
I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Thesis is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the thesis for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)

(Supervisor's Signature)

221-35-1009

Student ID

Date: 22/12/2025

Dr. Marzia Ahmed

Name of Supervisor

Date: 22.12.2025

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

APPROVAL

This thesis titled on "End To End YOLOv12 Based Multi Stage Pipeline for Bangla License Plate Recognition", submitted by **Iffat Ara Nowshin (ID: 221-35-1009)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Fazla Ealhe

Chairman

Dr. Fazla Ealhe
Assistant Professor & Associate Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Marzia

Internal Examiner 1

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Shabnom Mustary
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Md. Rajib Mia
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Mohammod Abul Kashem, PhD
Professor
Department of Computer Science and Engineering
DUET, Bangladesh

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

_____

(Supervisor's Signature)

Full Name : Dr. Marzia Ahmed

Position : Assistant Professor

Date : 22/12/2025

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

(Student's Signature)

Full Name : Iffat Ara Nowshin

ID Number : 221-35-1009

Date : 22-12-2025

ACKNOWLEDGEMENTS

All the praise worth Almighty Allah who granted me the strength and patience to do this thesis, from generating the idea of the topic to completing the study. Without his countless blessings and guidance, I would not be able to reach my goal.

My heartfelt thanks and gratitude go to my amazing parents, Md. Kowser Alam and Mst. Shamsunnahar Akter, for continuously supporting me, encouraging me the whole time and having faith on me. I truly acknowledge their love, support and sacrifices which give me the inspiration to study.

I am much obliged to my supervisor, Dr. Marzia Ahmed, Assistant Professor, Software Engineering Department, Daffodil International University, who guide me throughout my studies. Her guidance, supervision, and support in helping me bring out my best. Her assistance, feedback and constant encouragement have strengthened my knowledge in this topic and shaped the direction of the work.

Special thanks to Dr. Imran Mahmud sir , Head and Professor of the department of Software Engineering, Daffodil International University, for expanding our department day by day, emphasizing on the arrangement of modern high configured computer and labs, supportive academic environment and motivating us to achieve our goals.

DEDICATION

To my family, whose always inspire me with selfless love, continuous support and encouragement.

ABSTRACT

Automatic License Plate Recognition (ALPR) systems are critical for intelligent transportation and security infrastructure yet remain challenging for scripts with complex characters like Bangla. Bangla license plate contains complex structure such as curved glyphs, complex conjuncts and area-specific layouts which makes traditional OCR-based pipelines to struggle in the presence of occlusion, motion blur and low-resolution surveillance footage. This paper presents a novel multi-layer end-to-end Bangla ALPR system using YOLOv12's attention-centric architecture. The proposed pipeline utilized a lightweight family of YOLOv12 models so that it can make the feature representation more consistently optimized across vehicle, plate and character detection and improve robustness to scale variation of urban backgournd. We introduce a three-layer model approach: (1) a YOLOv12-based vehicle detection model (0.975 mAP@0.50, 0.924 mAP@0.50:0.95, 2.3 ms/inference), (2) a YOLOv12n license plate detection model (0.975 mAP@0.50, 2.3 ms/inference), and (3) a specialized YOLOv12 character recognizer for Bangla glyphs (0.986 mAP@0.50, 0.750 mAP@0.50:0.95), eliminating OCR dependencies. All the layers are trained on real images of Bangladeshi traffic scenes covering various illumination, cluttered urban scenes, diverse viewpoints and multiple plate layouts to ensure a generalized to real roads of Bangladesh. Trained on real-world Bangladeshi vehicle datasets, our system processes 640×640 resolution images on a consumer-grade GPU. The character recognition model handles 102 classes including conjuncts such as ক্ষ (kkho), জ্ঞ (gya) etc through coordinate-based reconstruction, achieving reliable detection and recognition of Bangla license plate numbers in unconstrained traffic scenes. This study proposes a fast and reliable Bangla license plate recognition solution for real-life traffic scenes and establishes a YOLOv12-based pipeline capable of complex-script ALPR.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

R	Set of real numbers.
$x \in R^{\{3 \times 640 \times 640\}}$	Input RGB image to the YOLOv12 network (3 channels, 640×640 resolution)
F, F'	Original feature map from backbone and attention-refined feature map
Q, K, V	Query, Key, and Value tensors used in the attention mechanism.
W_Q, W_K, W_V	Learnable weight matrices that project features into (Q, K, V).
d_k	Dimension of the key (and query) vectors in attention.
$\mathrm{Attn}(Q, K, V)$	Scaled dot-product attention output.
$\sigma(\ )$	Sigmoid activation function mapping values to ([0, 1]).
s_{ij}	Raw logits at spatial location ((i, j)) in the detection head.
$\hat{P}_{ij}$	Predicted class probability vector at location ((i, j)).
C	Number of object / character classes.
$\delta, y, n.$	Ground-truth offset y ; its integer bin $n = \lfloor y \rfloor$; and fractional part $\delta = y - n$.
pk	Predicted probability of bin k in distribution focal loss.
$\mathcal{L}_{box}, \mathcal{L}_{\mathrm{DFL}}, \mathcal{L}_{\mathrm{cls}}, \mathcal{L}$	Bounding box loss, distribution focal loss, classification loss, and total loss.
$\lambda_{box}, \lambda_{dfl}, \lambda_{cls}$	Weight coefficients for $L_{\mathrm{box}}, L_{\mathrm{DFL}}, L_{\mathrm{cls}}$ respectively.

LIST OF ABBREVIATIONS

ALPR	Automatic License Plate Recognition
LRP	License Plate Recognition
YOLO	You Only Look Once
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
DETR	Detection Transformer
CRNN	Convolutional Recurrent Neural Network
OCR	Optical Character Recognition
GPU	Graphics Processing Unit
NPU	Neural Processing Unit
IoU	Intersection over Union
NMS	Non-Maximum Suppression
DFL	Distribution Focal Loss
mAP	Mean Average Precision
AP	Average Precision
RGB	Red–Green–Blue color space
BGR	Blue–Green–Red color space
HSV	Hue–Saturation–Value color space
AMP	Automatic Mixed Precision
LPDB-A	A public Bangla license plate dataset
IR	Infrared
IoT	Internet of Things

CHAPTER 1

INTRODUCTION

1.1 Background

With the advancement of Computer Vision technology and its use in real-time object detection, has significantly contributed to mitigate real-world problems. One of its vital applications is automatic license plate recognition (ALPR) systems, which is crucial, especially for developing countries like Bangladesh, where traffic violations are so common. In addition, it has been more challenging to detect and recognize license plates with Bangla characters.

'YOLO (You Only Look Once)' series has dominated in recent years for its high speed, accuracy and efficiency [1]. While the previous models were based on CNN architecture, the newly introduced YOLOv12 came up with the concept of attention mechanism, which helps the model to recognize the patterns as fast as various models but more accurately [2]. YOLOv12 combines the mechanism of attention with the YOLO speed, which makes it both fast and accurate, and this the attention-centric framework refers to [3]. Despite these advances, no prior work adapts YOLOv12 for end-to-end Bangla LPR with optimized localization. In this paper, we propose a three-stage LPR system tailored for Bangla vehicles using YOLOv12 for vehicle detection, plate localization and character recognition, supported by a lightweight preprocessing pipeline and hyperparameter tuning.

1.2 Problem Statement

Despite continuous advancement, Bangla license plate recognition is still far too frequently unsuccessful in the wild. Although early CNN-based Bangla ALPR pipelines shown potential, they usually depended on strict post-processing that fails when it comes to real world scenarios and struggled across cameras, lighting, and angles or sometimes for dataset size and generalization limitations. Although multi-step deep models have enhanced detection and recognition, they are still vulnerable to plate inconsistency and picture noise, which caused mistakes in one stage to make their way down to the next. Another reason we addressed for system fail to detect plate correctly is when there is any other object, art or sign that looks like a number plate, model often get consumed and predict them as number plate [4]. Real-time systems reported good lab metrics, yet maintaining accuracy at edge speeds under rain, glare, motion blur, and tilt is still difficult, and end-to-end robustness across districts and fonts remains inconsistent. Even the newer ALPR systems tuned for Bangladesh struggle when the plate is not near-frontal, environment is messy (small, skewed, noisy Bangla plates) or low-res images. Current models may spot the plate accurately but it often miss or shuffle with complex Bangla forms and conjuncts as ক্ষ (kkho), জ্ঞ (gya) etc. Therefore coordinate-based reconstruction methods hardly yield precise plate numbers [5].

Moreover, most of the projects in Bangladesh faces challenge with tight budgets, unstable power, limited bandwidth, and heterogeneous cameras. Shipping video to the cloud or running heavy models on expensive hardware is often unrealistic. These are unignorable errors which not only effect the accuracy metrics but undermine trust in automated systems that could improve safety, accountability, and traffic management. To fix that, this thesis proposes two models: one for accurately locating the plate and another for quickly and accurately identifying each Bangla character or symbol with reasonable and within minimal maintenance.

1.3 Motivation

Our work initiates from the local need of Bangladesh's roadways, with their heavy traffic and linguistic diversity which creates a particularly difficult environment for automated analysis. However, existing recognition algorithms are still insufficient to reliably read Bangla script when rain, glare, fog, motion blur, or camera angle change, especially when conjunct characters appear. This problem was selected to find a solution that genuinely recognizes Bangla letters, numbers, district markers, and conjuncts, so that it can directly improve public services here.

Additionally, reducing the cost of deployment and maintenance was also priority. Our suggested model, which is based on YOLOv12 and has lightweight variations intended for edge devices, allows inference to run on commodity PCs as well as modest GPUs or NPUs, lowering both capital costs and continuing bandwidth/compute bills. Our goal is not only to recognize Bangla plates accurately but also to do so with a practical architecture that satisfies regional limitations, one that is inexpensive to implement, easy to use, and sustainable to maintain.

1.4 Significance of the Study

This study offers a reliable method for fast and accurate Bangla license plates on real world conditions. It has three separate model design to detect the vehicle region first, then the plate position with enhanced localization accuracy and another one to read Bangla character fluently with small, crowded glyphs and conjuncts. It reduces confusions with road signs, billboards and other objects which looks like e license plate in the image and improve precision in real deployments. The process is simple to train and implement, provides clear metrics and visuals which offers a useful baseline for future development.

1.5 Research Questions

In this study, we tried to find answers to the following questions:

- Would a YOLOv12 tri-model setup (first model to locate the vehicle region, second one to detect plate and last one to interpret characters) perform better than a single model approach with ORC integrated on Bangla license plates?
- How better does the character recognition model detect the small, crowded Bangla letters and conjuncts (e.g., ক্ষ, জ্ঞ) when pictures are low resolution, blurred or curved due to camera angle?
- Is it possible for the entire pipeline to deal in real time on affordable edge hardware such as small GPU or NPU or a normal PC, while maintaining accuracy in the face of motion blur, rain, and glare?

1.6 Research Objective

This study initiated with the following objectives:

- A tri-model YOLOv12 pipeline that distinguishably separates plate detection from character/symbol detection, improving accuracy on complex Bangla letters and words (including conjuncts) without compromising real-time speed.
- Lightweight YOLOv12 variants sized for edge devices (modest GPUs/NPUs or commodity PCs), lowering capital expense, bandwidth needs, and recurring compute costs.
- When districts or fonts change, decoupled stages reduce downtime and field retuning by enabling the update or retraining of a single model (e.g., characters).

1.7 Research Scope and Limitations

The following section acknowledges the scope of this study along with the limitations from the dataset, chosen techniques, and evaluation process.

1.7.1 Scope

- End-to-End Bangla LPR using YOLOv12-Based tri-Model with left-to-right reconstruction (Model-1 for vehicle detection, Model-2 for plate detection, Model-3 for character detection).
- Experiments were performed from a public Bangladeshi vehicle/plate dataset in JPG/PNG with YOLO annotations, 1928 vehicle images with visible Bangla plates(1class for plate) and 2662 cropped plate images where 720 synthetic and 1942 manually cropped with 102 classes (digits, letters, district markers, separators). No multilingual, vanity or nonstandard layouts are used.
- Supervised training with fixed train/val/test splits on YOLO format dataset, Standard detection metrics (Precision, Recall, mAP, F1) and inference speed (FPS/latency) used for evaluation.
- Instead of using sequence recognition models (CRNN/CTC), language models, or multi-frame methods, the character recognition model is treated as an object detection task in single RGB images.
- Designed for low-cost edge hardware like small GPU/NPU or a normal PC for typical Bangladesh road scenes, excluding cloud servers, custom chips (FPGA/ASIC), and phone-only configurations.

1.7.2 Limitations

- Data has been collected, train and test mainly on Bangla LPDB-A. It may not represent other cities, new camera variations, unseen plate styles/font or night-only IR footage.
- The model is limited to single RGB frames. The pipeline doesn't use multi-frame fusion or video tracking, as a result extremely blurry or heavily obstructed plates might remain hard.
- Left-to-right coordinate sorting is used for reconstruction. This may misorder near characters and be less reliable on angled, stacked, or partially visible plates.
- Sequence recognizers (such as CRNN/CTC or language models) were not included which might be beneficial for some difficult cases like complicated conjuncts on extremely low-resolution crops but out of scope here.

- The pipeline is adjusted for affordable edge devices to reduce cost but in comparison to large, server-grade models, it also limits model size and may limit accuracy.
- Report standard metrics (Precision/Recall/mAP, F1, FPS) used in a controlled test environment setup. Large-scale operational KPIs, complete enforcement workflows, and long-term field experiments are not included.

1.8 Thesis Organization

This thesis contains five main chapters, closing with references and appendices. The first chapter is the introduction where the background of the proposed model described, addressing the problem, explaining why it matters and defines the scope and limits. The second chapter reviews related works done on license plate detection and recognition and addresses the research gaps that still remains. The third chapter explains the methodology of the proposed dual-model YOLOv12 system, how the dataset pre-processed, architecture of selected models, and a detail process from plate detection to number extraction. It also explains the training settings and the metrics for evaluation. The fourth chapter shows the results, compares models, report accuracy and speed, provide examples and examine typical mistakes such conjuncts and tiny glyphs. The last chapter contains the key findings, contributions, reflect on limits and suggest for further improvements. All cited works are listed in the references, and additional figures, configurations, and other supporting information are included in the appendices.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Works

The development of Automatic License Plate Recognition (ALPR) systems adapted for Bangla license plates has been a key research priority, especially due to the inherent complexity of the Bangla alphabet, variable plate designs, and diverse environmental circumstances in Bangladesh [6]. The main approaches used in the initial Bangla ALPR studies were conventional image processing and machine learning techniques. Saif et al. [7] used Convolutional Neural Networks (CNNs) in their system and reported a high accuracy of 99.5%. Shomee et al. [8] extended this study by applying a multi-step deep learning model for a variety of Bangladeshi license plates, achieving a Mean Average Precision (mAP) of 98.09% for character recognition and 98.35% for plate detection. Besides these, BLPnet used a cascaded DNN architecture with a dedicated Bengali OCR engine, by reducing computational cost for real-time use and emphasizing rotation-invariant character recognition for end-to-end plate detection and recognition [9]. A two-stage detection with lightweight backbones can achieve acceptable accuracy on embedded platforms across a real-time Bangla license plate pipeline for low-resource video applications further demonstrated [10]. A robust study in most recent time used CNN- and GAN-based restoration pipelines have been explored to handle blurred, low-resolution Bangla plates shown improved robustness under degraded imaging conditions using GFP-GAN followed by CNN character recognition [11].

The development of efficient deep learning-based object detection has brought about a significant change in the ALPR system. Using Faster R-CNN for the first time , Mahmood et al. proposed a dual layer license plate detection framework to detect vehicles and crop license plates which applies HSV color processing, morphological filtering and then size constraints inside the vehicle region [12]. This study addresses how restricting the search to analyze vehicles' location might improve the result for plate detection but it still uses manually created parameters for plate localization. The "You Only Look Once"

(YOLO) series was first presented by Redmon et al. [13], and real-time applications have been built on top of its single-pass detection concept. ALPR activities have gradually integrated with YOLO variants. For the Bangla ALPR, Tusar et al. [14] found a 78% character recognition rate with EasyOCR (2022) and a 98% detection accuracy with YOLOv5. The other Bangla-focused studies emphasized enhancement and detection together such as diverse-quality image pipelines that target low-light, blurred, and low-resolution Bangla plates using dedicated recognition strategies [15] and fog-resilient car plate recognition that uses Dark Channel Prior–based dehazing followed by YOLO detection to improve performance in adverse weather [16],

YOLOv8 has become more popular in this field more recently; in 2024 [17] presented a YOLOv8-based system that achieved a mAP of 96.3% for Bangla license plates. Another journal study that integrated YOLOv8 on a selective web dataset of 270 photos with image processing and OCR claimed over 99% plate detection accuracy and approximately 98% character recognition, offering a solid baseline for detection and reading pipelines [18]. A few more Bangla-focused studies used YOLOv8 to detect license plate and EasyOCR or custom CNNs to recognize letters: One IJRAS paper suggests a YOLOv8 + Roboflow workflow specifically designed for moving vehicles, angular views, and partially broken plates in actual Bangladeshi scene s [19], while another Authorea preprint reports enhanced plate localization and Bangla s cript recognition using YOLOv8 with Roboflow-based augmentation on CCTV-like images [20]. A dual-component VLPR system from Bangladesh emphasizes real-time traffic surveillance applications by combining a proprietary CNN for character rec ognition with YOLOv8 for plate detection [21].
Separate studies concentrate on YOLOv8based Bangla plate detection, training speciali zed detectors for precise plaque localization in parking and surveillance scenarios [22]. Beyond Bangladesh, streamlined YOLOv8 pipelines have been suggested for resource-constrained devices such low-power embedded boards [23], and improved YOLOv8-based ANPR systems have been built for complicated multi-format plates (e.g., in Qatar) employing dataset augmentation and edge computing [24]. Recent conference and journal articles have also described generic YOLOv8+EasyOCR pipelines as useful end-to-end templates for real-time license plate recognition and reading [25].

Beyond fundamental YOLO improvements, researchers are exploring ways to improve these models; for example, attention processes were added into YOLOv8 for Chinese license plate identification to improve feature focus [26]. In order to handle the complexities of the pattern, CRNN-based methods have also been studied for Bangla characters in the character recognition component [27]. Even with these developments, achieving even greater localization accuracy for license plates remains a significant issue for reliable, real-time deployment, especially when dealing with the complex Bangla characters and the frequently changing environment of Bangladesh. A YOLOv9-based Bangla license plate system that assesses several YOLOv9 variants and reports th e best F1 score above 0.95 on a multi-city dataset is one example of recent Bangla-oriented research that also looks into robustness to domain shifts (weather, low quality, diverse cities) and provides comparative analyses across YOLO versions [28].

Developments in loss functions, including Distance-IoU (DIoU) [29] and Focal-EIoU [30], have been widely used in general object identification to push the limits of localization precision since bounding box regression is much enhanced by these improvements, which is crucial for precise plate detection and the character identification. This research implements YOLOv12, the latest version of the YOLO architecture, to advance Bangla ALPR. Real-time object detection has advanced significantly with YOLOv12's groundbreaking attention-centric feature fusion [31]. These built-in enhancements are intended to increase general robustness and object detection accuracy. We can see the glance of it in the recent studies, such as a attention-centric study in 2025 reports mAP@0.5 around 0.996 and .994 on CCPD and an Indian dataset which is useful for comparing attention solutions with YOLOv12 and analyzing robustness under challenging settings [32]. Another study in the same year used improved YOLOv8n models with updated blocks and WIoU loss increased accuracy (mAP@05) from 90.9% to 94.4% and achieved about 86 FPS on a surveillance plate dataset, which makes them real-time edge application [33].

Table 2.1 Existing Models

Year	Dataset	Models	Results Accuracy	Cite
2022	Custom Bangla license plate dataset (≈295 images; Kaggle listing tied to the paper)	"MD-YOLO" (real-time pipeline)	IoU 99.5% and accuracy ≈0.9951 reported in the paper.	11
2024	270 images collected from the web (annotated with CVAT)	YOLOv8 (plate detection) + image processing + OCR	Detection accuracy 99%; character recognition accuracy 98%.	13
2024	Dataset derived from Hossain et al. (localization ≈2,800 imgs; character ≈4,000 imgs; converted to YOLO, split 80/10/10)	YOLOv8 variants (n/s/m/l/x)	Best: YOLOv8-X mAP50 = 0.96, mAP50-95 = 0.75.	12
2025	Custom surveillance LP dataset (varied lighting, backgrounds, angles, vehicle types)	Improved YOLOv8n (modified C2f, SPPF, head; WIoU loss)	mAP@0.5 ↑ from 90.9% → 94.4%; Precision 92.8%; Recall 87.9%; 86 FPS.	20
2025	CCPD and Indian Vehicle Datasets	YOLOv12-based detector with SimAM + Coordinate Attention; anomaly-aware augmentation via Latent Diffusion.	mAP@0.5 = 0.996 (CCPD) and 0.9937 (Indian); mAP@0.5:0.95 = 0.9108; Precision 0.9831; Recall 0.9747.	19
2025	(General object detection; standard real-time benchmarks)	YOLOv12 (attention-centric)	YOLOv12-N: 40.6% mAP with 1.64 ms latency on T4; outperforms YOLOv10-N/YOLOv11-N at similar speed.	3
2025	Custom Bangladeshi LP dataset (≈2,600 images; varied lighting/weather; EasyOCR for Bangla)	YOLOv8 + EasyOCR	Detection accuracy 94.8%; OCR character accuracy 92.7%.	5

2.2 Research Gap

Although Bangla license plate detection (LPD) and recognition have improved in recent years, several important limitations remain. First, most existing Bangla ALPR studies focus on either plate localization or character recognition, rather than designing a single end-to-end pipeline that jointly handles vehicle detection, plate localization, and character recognition. As a result, intermediate errors (e.g., incorrect vehicle bounding boxes or missed plates) are not propagated and evaluated across the full system.

Second, many works still rely on traditional image processing, segmentation, and generic OCR engines that are mainly tuned for Latin characters. These pipelines struggle when plates appear with noisy backgrounds, motion blur, partial occlusion, low resolution, and strong illumination changes, and they often fail to recognize complex Bangla characters, conjuncts, and mixed letter–digit layouts.

Third, deep learning-based methods that have been proposed for Bangla plates typically use small or limited datasets, often covering only digits or a subset of Bangla characters, and are evaluated on constrained or manually captured images rather than unconstrained traffic scenes. This makes it difficult to judge their robustness in real Bangladeshi road environments. In addition, many studies do not report detailed real-time performance (inference latency, FPS) on consumer-grade GPUs or edge devices, so their practical feasibility for deployment remains unclear.

To address these gaps, this study proposes a multi-layer YOLOv12-based lightweight ALPR system for Bangla license plates that performs vehicle, plate, and character detection in an integrated end-to-end pipeline, is trained on real-world Bangladeshi traffic data with complex backgrounds and plate layouts, and reports both accuracy and runtime performance on consumer-grade hardware.

CHAPTER 3

METHODOLOGY

We construct our Bangla license plate detect system by following a straightforward end to end workflow. At first, the images are collected from a public dataset Bangla LPDB-A dataset on Zenodo (single vehicle images with visible plates and cropped plate images). While verifying the labels and classes of the plate, some wrong classes was found. So we manually corrected all the labels and classes that we found during review. YOLOv12 has a by default preprocessing set up which automatically handle some basic steps such as read and convert images into a common size (e.g. 640) while keeping the aspect ratio with padding, pixel values are normalized, default augmentation are applied during training , label files in YOLO format and train, test, validation splits can be created from the data file [34]. Some manual preprocessing still done to ensure that all the annotation truly matches it's image or not, their formats and correspond properly to the images, the quality of the images (e.g., lighting, size, aspect) and the split prevents data leaking across similar kind of images. Additional augmentations are added for domain specific in order to help Bangla letters and marks, such as light random crops, little rotations and tilt, slightly changes in brightness and contrast. To align the scaling and padding with the dataset, a suitable image size and other training settings are selected. Then the first model is train to detect the plate from full vehicle images and it returns coordinates as array of the detected part and these are cropped accordingly to pass into the second model that is trained on the cropped plates with a 102 class which includes digits, Bangla letters, vowel marks, conjuncts and separators and it returns small boxes for each characters and marks of the plate. A basic post-step arranges detections from left to right. Translates class ids to Unicode and joins combines close parts to rebuild conjuncts which generates the final plate string. Lastly the entire system is tested on the help out set and also from some random images clicked with multiple vehicles in single frame and then the accuracy of plate detection, character detection , end-to-end string are reported. This proposed models are packed for minimal cost edge use on a general GPU or NPU or any normal configured PC for practical deployment in Bangladesh.

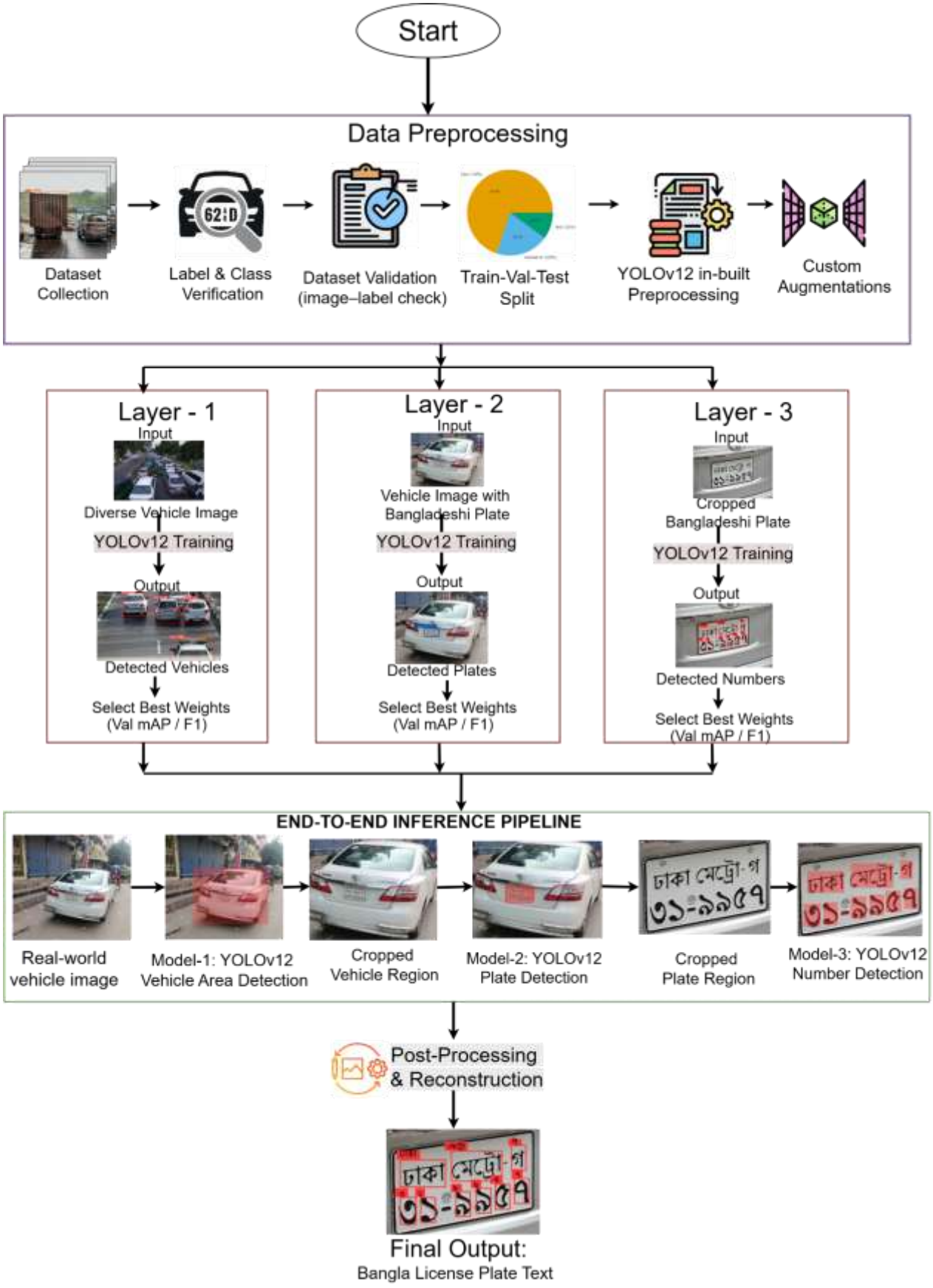


Figure 3.1 Step by step procedure diagram

3.1 Data Collection

For vehicle image collection, a publicly available dataset where there was over 3000 images of different vehicles with annotations and for plate localization and number recognition , we used a different public dataset of Bangla LPDB-A dataset via Zenodo [35]. This dataset includes varied real-world conditions (e.g., daylight, shadow, partial occlusion) and diverse vehicle types. The dataset has been split into two parts , one containing full vehicale images with single class (plate) and other one containing the licence plates that have been cropped and annotated with 105 classes including digits, Bangla letters, conjunct parts and separators. Later we just kept 102 classes as there was some duplicate classes. The dataset was manually filtered and relabeled in the YOLO format. Each image has an accompanying .txt annotation file in the format:

<class_id> <x_center> <y_center> <width> <height>

All bounding box coordinates are normalized between 0 and 1. The dataset was split into training and validation sets using an 80:20 ratio .

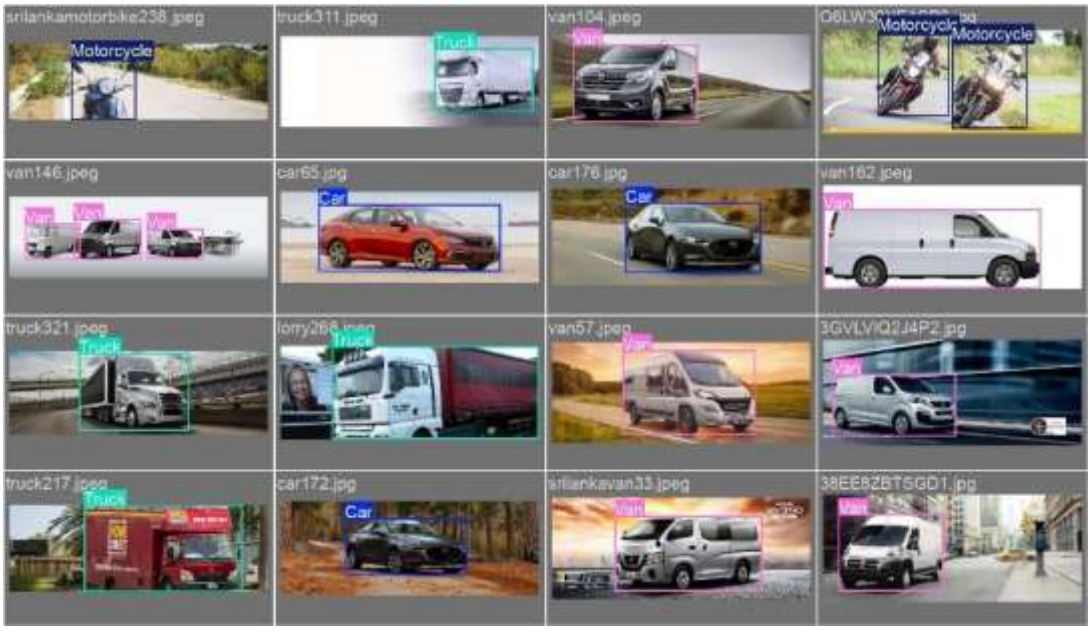


Figure 3.2 Sample annotated vehicle images

Figure 3.3 Sample annotated plates in vehicle images

Figure 3.4 Sample annotated number in cropped plate images

3.2 Data Preprocessing

Data preprocessing is one of the vital step in model training which not only reduce confusion but also accelerates the model accuracy and speeds up learning. That's why it's always been a top priority for the Data Scientists. Though YOLOv12 comes with various in-built preprocessing including reading input images and converting to RGB, resizing images, normalization of pixel values, splitting data into train/test/validation, augmenting, bounding box labelling and many other. But there are still some parts which need to be check manually. The following preprocessing steps are taken for the proposed model.

3.2.1 Label Verification and Correction

Correctness of the annotation is very sensitive for Model training. Even though the dataset had YOLO-format annotations in itself, a careful review uncovered several inconsistencies including incorrect classification, duplicate labels, misaligned bounding boxes and some missing characters on the cropped plates. To ensure expected accuracy, every annotation file was reviewed individually with its corresponding image. Incorrect labels were fixed; mislabelled characters were reallocated to their correct classes and bounding boxes were modified to capture target area.

3.2.2 Removal of Duplicate and Redundant Classes

During dataset investigation, some classes appeared more than once in 105 classes and the have allocated different classes for duplicates. This misclassification can confused model and reduce the model accuracy. Thus, all duplicate classes were removed and found 102 class in total. Removal of duplication and redundant classes helps character detection model to be clear and unambiguous.

3.2.3 Image-Label Consistency Checking

Before model training, checking image-label consistency assure the structural consistency. The bounding boxes were within valid normalized ranges of 0-1, verifying images has accurate corresponding annotation file and ensuring each annotation files has

correct number of lines within it – are included in this check. Annotation files with inconsistent formats, unclear labels and missing annotations are fixed or removed.

3.2.4 Enhancing Image Quality

Noisy samples like extremely low-resolution images, partially blurred images, improper cropping or plates are partially hidden to recognition – are common for real-world datasets. The models get confused by these type of data often. Therefore, the image quality in the dataset were manually reviewed and either eliminated or re-annotated when feasible. During the model training , this selective filtering would be beneficial to reduce annotation noise, produce more consistent convergence and outputs are more stable.

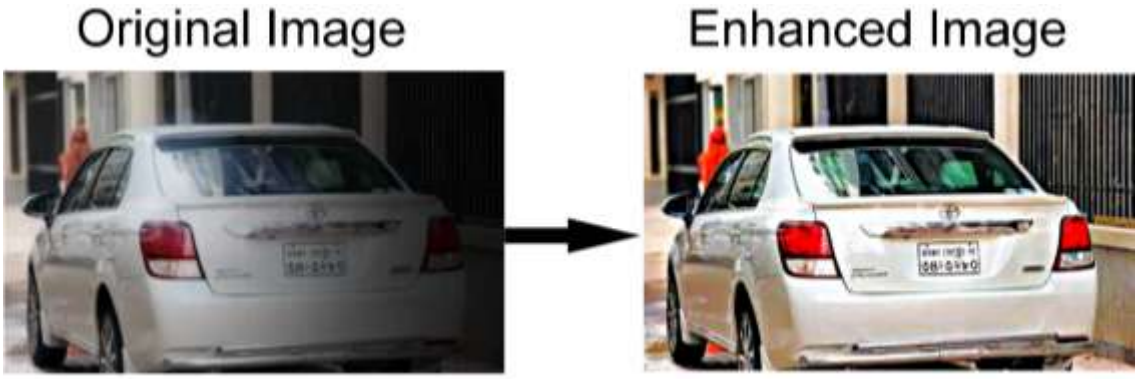


Figure 3.3 Applying custom correction to enhance image quality

3.2.5 Data Splitting

The dataset is divided into train-validation-test subsets in the following portion: 70 : 20 : 10 to avoid data leakage and maintain the evaluation process in a proper manner. Since the dataset have both cropped plate images and full vehicle images, cautious inspection was required to ensure that visually similar images did not appear in multiple segments. Dataset was thoroughly inspect to ensure that the plate crops from the same source image of the same vehicle reminded within a single subset for both before and after splitting. This part also ensured that each image had it's corresponding annotation file after partitioning which helps to prevent mismatches that could affect model training.

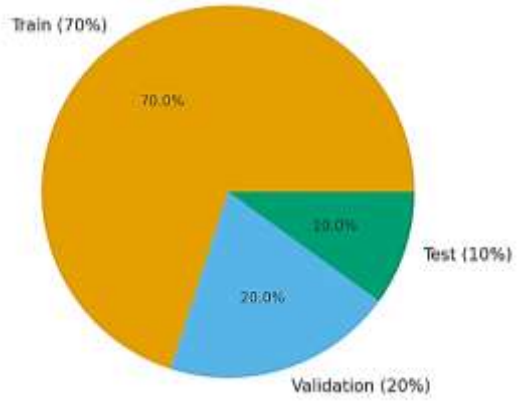


Figure 3.4 Dataset splitting portions

3.2.6 Standardization of Image Dimensions

YOLOv12 comes with in-built data preprocessing feature which includes automatic image resizing to 640x640 pixel if the dataset contains images with different aspect ratios and resolutions. Images with unusual aspect ratios were reviewed individually and minimized excessive padding or anomalies to ensure a consistent structure during automated scaling. This step helps to enhance the effectiveness of both detection phase and enable YOLOV12 in maintaining stronger spatial connections between characters.

3.3 Parameters and hyperparameters

Our proposed end-to-end system used YOLOv12 object detection framework and the same optimization set up used to train for vehicle detection, plate localization and character recognition models. YOLOv12 have predefined architectural parameters with integrated training pipeline, decoupled head for multi-scale detection and task-specific loss functions suited for multi class detection and bounding box regression. Each models were trained for 500 epochs with batch size of 32, applying the auto-optimizer section of YOLOv12 which usually corresponds to an SGD/AdamW version depending on the computing device. The training used automatic mixed precision (AMP) to speed up computation and the initial learning rate was set to 0.01 with an equivalent final learning

rate. Momentum was set at 0.937 and weight decay rate was set at 0.0005 in accordance with YOLOv12's recommended settings for stable convergence.

The loss function includes three components, bounding-box regression loss (weight = 7.5), classification loss (weight = 0.5), and distribution focal loss (DFL, weight = 1.5), which combinedly assist the mode in learning object location, category prediction and high-resolution bounding box distribution. To maintain early gradient updates, momentum – 0.8 was used during a warmup phase of 3 epochs. Data augmentation techniques during training included horizontal flipping, HSV shifts, mosaic augmentation, copy-paste mixing and controlled random erasing. To preserve character shape integrity, geometric distortion parameters such as rotation, shear, perspective were maintained to a minimum. A confidence threshold of 0.25 and IoU threshold of 0.45 were applied during the inference of non-maximum suppression (NMS) module.

Table 3.3.1 List of Parameters used in all Models

Category	Parameter	Value
Training	Epochs	500
Training	Batch Size	32
Learning Rate	Initial LR	0.01
Optimizer	Momentum	0.937
Loss	Box Loss	7.5
Loss	Class Loss	0.5
Loss	DFL Loss	1.5
Inference	Confidence Threshold	0.25
Inference	IoU Threshold	0.45

3.4 Model Selection

3.4.1 YOLOv12

YOLO versions were using CNN before the latest version of its' family "YOLOv12" which introduced the concept of attention mechanism. YOLOv12 combines the power of attention blocks inside the backbone and neck with YOLO speed which makes it bpth fast and more precise. Using this , it can con focus on relevant areas only and ignore the background. As a result it can outperform where the target items are small or packed such as Bangla characters, vowel marks, and conjunct portions in license plate.

YOLOv12 improved performance for handling bounding box for small objects. YOLOv12 uses a distribution-based prediction, instead of predicting only one value for each box side which gives more precise control over box positions. This helps the model create precise boxes around small characters or objects. Since Bangla license plates contains complex characters, conjunctions, vowel marks etc, a model which have stronger feature focus on small details while keeping similar or better speed is needed for better result. For this, you chose YOLOv12 over YOLOv8 or YOLOv4. All the properties of YOLOv12 suits Bangla license plate recognition.

3.4.2 Faster R-CNN

Faster R-CNN is known as One of the most significant two-stage object identification architectures which frequently employed as a solid foundation in studies. Faster R-CNN works in 2 stages: Region Proposal → Classification pipeline. In the Region Proposal Network (RPN) stage, is scans the feature maps to create a set of potential bounding boxes known as region proposals. Then is the next stage (Classification), it uses a dedicated detection head to improve each proposal and classify it into object categories. By following this two-stage mechanism, Faster R-CNN achieves high localization accuracy, particularly for medium-sized objects and makes it historically one of the most reliable detectors in tasks requiring precise bounding boxes.

However, Faster R-CNN is significantly slower compared to modern one-stage detectors like YOLO due to additional computational overhead caused by the multi-step

inference pipeline. real-time performance is typically challenging to achieve without significant model compression or hardware acceleration. The architectural design of Faster R-CNN contrasts strongly with YOLO's real-time single-stage approach. This provides a opportunity to assess the efficacy of modern license plate identification compares with an older but well-established detection framework. The accuracy, robustness, and speed can be directly compared by training Faster R-CNN on the same Bangla license plate dataset.

3.4.3 DETR

DETR (recognition TRansformer) offers an entirely new approach to object recognition by replacing a transformer-based, anchor-free, set-prediction formulation for convolution-based region proposal mechanisms. DETR works based on transformer encoder–decoder architecture by modeling global image relationships and directly predicts a fixed set of objects. This substitute elements like anchor design and non-maximum suppression (NMS), providing a conceptually simpler and end-to-end trainable detection pipeline. One of the main advantages of DETR is its global receptive field which enables it to detect long-range dependencies and contextual cues that conventional CNN-based detectors may miss. However, the original DETR architecture often struggles to handle small and closely spaced objects due to its poor feature precision and slow training convergence. But this type of limitations are especially relevant to Bangla ALPR which have tightly spaced characters and small text structures often. In this study, Faster R-CNN is used as a modern transformer-based comparison model which reflect the current evolution of object detection studies. Comparing DETR with YOLOv12 and Faster R-CNN provides a comprehensive analysis on three major detection paradigms such as single-stage YOLO, two-stage CNN (Faster R-CNN), and transformer-based detection (DETR). This comparison assess the performance of transformer-based architectures on detailing tasks like license plate localization and determines whether their global reasoning capabilities can make up for their shortcomings in small-object detection.

3.5 Model Training

3.5.1 Vehicle Detection

The first stage of the pipeline uses YOLOv12 to detect vehicle from unconstrained different types of vehicle images like car, van, truck and motorcycle. The images were previously annotated with bounding box surrounding the vehicle region tightly. Because of this tightly enclosed bounding box, model can be train and learn where the vehicle exactly is and detect accurately. Some preprocessing steps of YOLOv12 such as images are resized, standard augmentations are applied such as mosaic composition, horizontal flipping, random scaling and HSV color jitter which makes the model more adaptable to different real traffic scenes like lighting, background, various viewpoints and scale.

The model outputs a set of predicted vehicle bounding boxes with confidence scores. To gain only most confident boxes and remove overlapping or duplicate detections , Non-Maximum Suppression (NMS) with an IoU threshold of 0.5 is applied for each vehicle. The detected area of a vehicle is cropped and used as input for next stage. It helps to reduces false positives of the system from background text regions, signboards or other plate-like structures in the image.

Figure 3.5 Sampl Output of Vehicle Detection Model

3.5.2 License Plate Localization

As the first model, this stage uses YOLOv12 to detect the license plate from the a bunch of vehicle image dataset. The dataset consists of annotated vehicle images with bounding boxes tightly enclosing the license plate region. Data augmentation techniques such as random scaling, horizontal flipping, mosaic augmentation, and HSV color jitter are applied to improve robustness to illumination, angle variation, and background clutter.

The detection head produces bounding boxes for the license plate class, which are post-processed using Non-Maximum Suppression (NMS) with an IoU threshold of 0.5 to remove redundant detections. The highest-confidence detection is selected as the final plate location. This bounding box is then used to crop the plate region from the original image for the next stage.

Figure 3.6 Sample Output of Plate Localization Model

3.5.3 Character Recognition

The second YOLOv12 model is trained specifically for Bangla license plate characters. The dataset comprises cropped plates with annotated bounding boxes around each individual character, including digits (০–৯), letters (e.g., ক, ল, র), and the dash symbol. The character detector predicts multiple bounding boxes per plate image. The output detections are sorted left-to-right based on bounding box xxx-coordinates to maintain correct reading order. Each class ID is then mapped to its corresponding Bangla Unicode character using a predefined lookup table. Finally, the sequence of recognized

characters is formatted according to the Bangla license plate standard, having Four-Digit Serial −

"City - Vehicle class letter - Registration series - Vehicle number"

The formatted string is overlaid on the original vehicle image along with the detected plate bounding box to generate the final annotated output.

Figure 3.7 Sample Output of Number Recognition Model

3.5.4 Full Pipeline

Our proposed model is based on attention-centric one-stage object detector which maintains the traditional backbone-neck-head structure of YOLO which is foundation of the suggested Bangla license plate recognition system, but it substitutes many pure convolution blocks with efficient attention blocks. This architecture used in both model, first model detects the vehicle present in the image. Then the license plate from the vehicle region, and third model recognize 102 classes of characters and symbols within the cropped plates. After normalization and scaling, let the input image be

$$x \in R^{\{3 \times 640 \times 640\}} \quad (1)$$

where x is input RGB image and R is the set of all real numbers.

At first the images are run though standard preprocessing (resize, normalization, padding), after that images are passes through the YOLOv12 backbone. The main components of this backbone are convolutional clocks and special attention blocks such as Area Attention and R-ELAN blocks which helps the model focus on relevant regions while maintaining computational efficiency. The attention block initiate project features

into query, key and value tensors for a feature map $F \in R^{C\times H\times W}$ and then YOLOv12 applies scaled dot-production attention on each local area of the feature map and then using residual connection, tis dot-production is added

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

$$Q = FW_Q \quad (3)$$

$$K = FW_K \quad (4)$$

$$V = FW_V \quad (5)$$

Where

W_Q, W_K, W_V = learnable weight matrices

Q , K , V = query, key, and value tensors

d_k = Dimention of each key vector

Using a residual connection, the output of the attention block is added back to the initial features:

$$F' = F + \mathrm{Attn}(Q, K, V) \quad (6)$$

Where

F = Updated feature map

F is the original feature map and residual sum allows the model to keep both the attention-refined data and original data which leads to stable training and improved feature reuse.

Figure 3.8 Attention Mechanism Visualization on Plane detection

The backbone produces several multi-scale features by stacking these blocks at various resolutions. There are sent to the neck which utilizes up-sampling , down-sampling and additional attention-aware blocks to merge data from coarse and fine scales. This combination plays a crucial role in out system since the Model-1 must identify plates that may appear at different sizes and Model-2 must detect small Bangla characters inside detected plates.

The YOLOv12 detection head doesn't have anchor. The head predicts tow types of output for each scale level and each spatial location (i,j) in the feature map. First one is a vector of class scores $s_{l,ij}$ and second one is a set of probability distributions that describe the distance from the four sides of the bounding box to the location using Distribution Focal Loss (DFL). Sigmoid function is applied to obtain class probabilities.

$$\hat{P}_{l,ij} = \sigma(s_{l,ij}) \tag{7}$$

where,

$\hat{P}_{ij}$ = Vector of predicted class probabilities at position (i,j)

s_{ij}= Correspinding vector of raw logits

σ ( ) = Element-wise sigmoid function

In our proposed plate detector (Model-1) , the number of C = 1 and C = 102 for the character detector (Model-2). YOLOv12 uses Distribution Focal Loss (DFL) for bounding boxes. It predicts a discrete distribution over K bins, instead of predicting a single continuous value for each box side. Let y be the ground-truth offset (assuming the distance in the feature units to the left side of the box) , we define :

$$\mathrm{n} = |y| \tag{8}$$

$$\delta = \mathrm{y} - \mathrm{n} \tag{9}$$

and let $\mathbf{p} = (p_0, \dots, p_{K-1})$be the predicted probabilities over the K bins. The DFL term for that side is

$$\mathcal{L}_{DFL}(y, p) = -[(1-\delta) log_{p_n} + \delta log_{p_{n+1}}] \tag{10}$$

Where

n = index of the lower bin

δ = the fractional part of the target offset

$p_n,\ p_{n+1}$ = the predicted probabilities of the two nearest bins

$\mathcal{L}_{DFL}$ = small when probability mass is placed close to the true offset

The model is encouraged to predict precise and accurate distribution for each box edge in order to localize extremely small characters on the plate. Weighted sum of box loss, classification loss and distribution loss determine the overall training loss for YOLOv12:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{dfl}\mathcal{L}_{\text{DFL}} + \lambda_{cls}\mathcal{L}_{\text{cls}} \quad (11)$$

Where

$\mathcal{L}_{box}$ = an IoU-based loss

$\mathcal{L}_{\text{DFL}}$= the distribution focal loss

$\mathcal{L}_{\text{cls}}$ = a binary cross-entropy loss

λ_{box}, λ_{dfl} , λ_{cls} = constant weights

For comparing each predicted box with its corresponding ground-truth box , $\mathcal{L}_{box}$ is used. $\mathcal{L}_{\text{DFL}}$ combines all four sides whereas binary cross-entropy loss between estimated class probabilities and ground-truth labels and all positive locations is measured by $\mathcal{L}_{\text{cls}}$. Constant weights (λ_{box}, λ_{dfl} , λ_{cls}) regulates the relative significance of localization, distribution, and classification terms during training.

YOLOv12 applies non-maximum suppression with an IoU threshold of 0.45 to remove duplicates and combines predictions from all feature scales, filters out low-confidence boxes using a confidence threshold (0.25 in our setup) during inference. One or multiple bounding boxes are created by this in Model-1. According to these bounding boxes, plates are cropped and passed to Model-2 which uses the same process of YOLOv12 to predict small boxes with 102 class IDs for Bangla letters, digits, vowel marks , conjunct parts, district names and separator. The proposed dual YOLOv12 system can precisely detect both the plate and small internal characters because of the backbone's attention and head's DFL-based box regression. This is crucial for correct reconstruction of the Bangla license plate in the later post-processing stage.

3.6 Evaluation Matrix

We used IoU-based metrics—precision, Recall, F1-score, Average Precision(AP) and mean Average Precision(mAP) for both plate detection (Model-1) and character detection (Model-2) as these are the most used evaluation matric in recent times. IoU with the ground-truth box is computed for each predicted box. Correct class labels and IoU over a threshold (e.g. 0.5) is considered as True Positive (TP); missed objects are False Positive (FN) and unmatched predictions are False Positives (FP). Precision represents the number of accurate predictions among all positive whereas Recall tells the number of objects found and F1 is the harmonic mean of precision and recall which useful for balancing them.

Precision-Recall curve is generated by adjusting the confidence threshold and the area under this curve indicates its AP. Average of AP for all classes is referred as mAP. We used two standard versions: mAP@05 and mAP@0.5:0.95 . Finally, for end-to-end Bangla license plate pipeline, we used plate-level accuracy.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{13}$$

$$AP = \sum_n (R_n - R_{n-1})\, P_n \tag{14}$$

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c \tag{15}$$

CHAPTER 4

RESULTS

4.1 Result Analysis

This section includes all the results from different models, their comparative study and how far the result improved by implementing 3 stage model is discussed. For comparative study, three different models were selected: (1) YOLOv12, (2) Faster R-CNN and (3) DETR. Standard evaluation metrics such as precision, recall, F1-score, mAP@0.5, mAP@0.5:95 are used. This study not only compares YOLO with other models but also represents how adding an extra layer to detect vehicle can helps the next models to improve in result.

4.1.1 Vehicle Detection Model

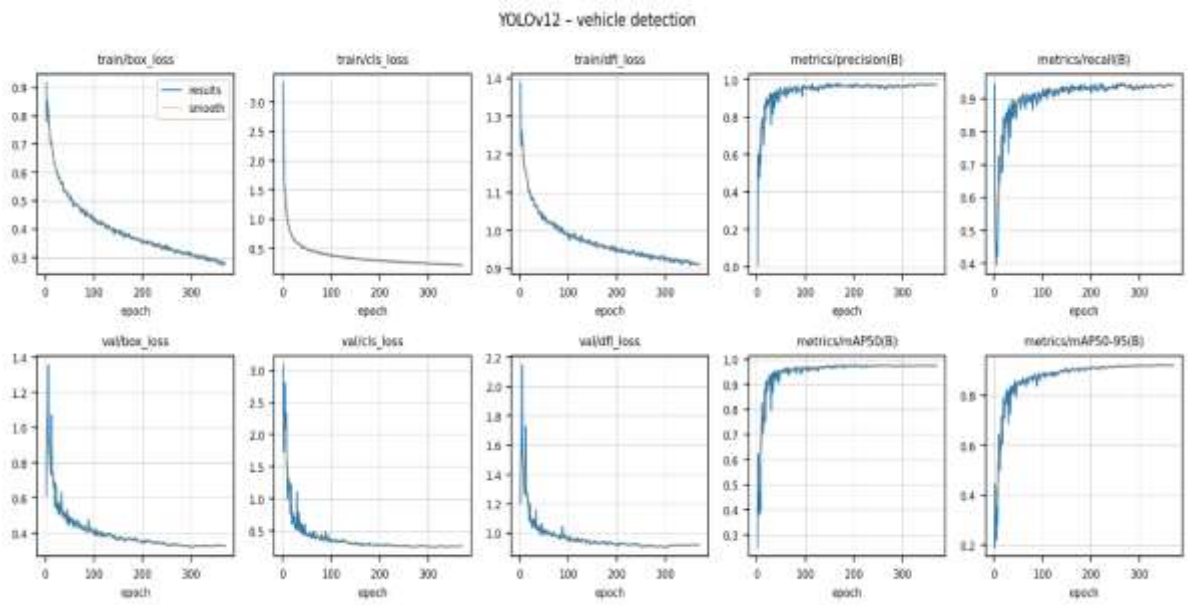


Figure 4.1 Training Metrics of the Vehicle Detection Model using YOLOv12

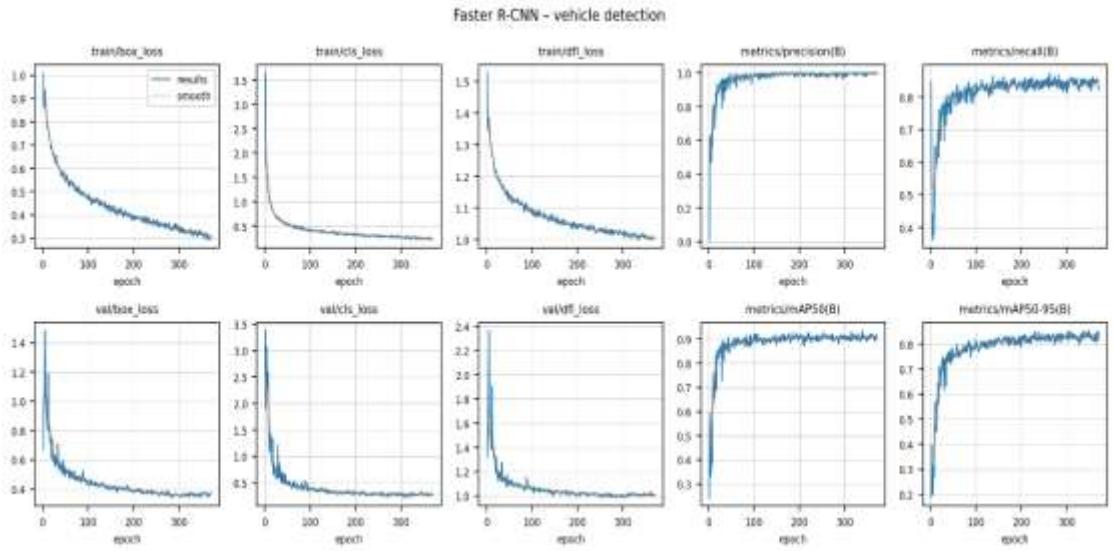


Figure 4.2 Training Metrics of the Vehicle Detection Model using Faster R-CNN

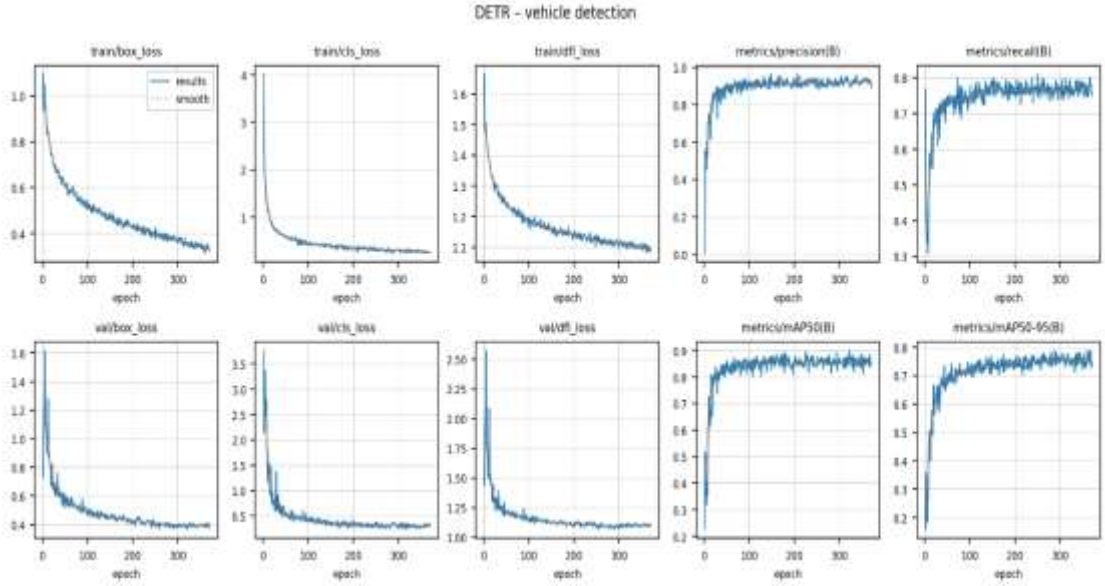


Figure 4.3 Training Metrics of the Vehicle Detection Model using DETR

Table 4.1 Overall Result Comparision of Vehicle Detection Model

Metric	YOLOv12	Faster R-CNN	DETR
Best epoch	210	338	322
Precision (P)	0.969	0.972	0.919
Recall (R)	0.945	0.851	0.791
mAP@0.50	0.978	0.935	0.899
mAP@0.50:0.95	0.913	0.846	0.755

Vehicle detection models show a decent result with training, validation losses decreasing and performance metrics improvement over epochs. The result shows that YOLOv12 reached 338 epoch to achieve the best checkpoints like precision of 0.976, a recall of 0.939 and an mAP@0.5 and mAP@0.5:0.95 of 0.975 and 0.924, respectively, which indicates accurate localization of vehicles. Whereas other two models likely close to this result but didn't cross the numbers. From normalized confusion metrics of YOLOv12 (figure 4.4), most of the vehicles are correctly detected by the models with class-wise accuracy around or above 90 but there is some limited confusion with the background class. This proves that YOLOv12 has a high accuracy to detect vehicle region.

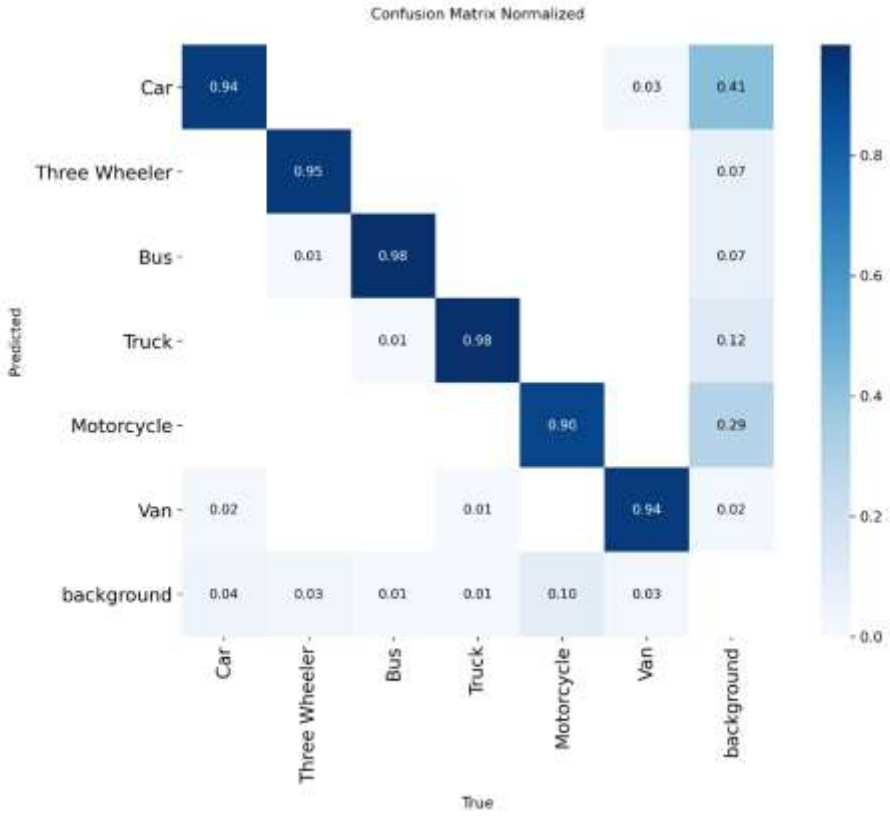


Figure 4.4 Normalized Confusion Metrics of YOLOv12

4.1.2 License Plate Detection Model

Table 4.2 Result Comparision Table of License Plate Detection

Metric	YOLOv12	Faster R-CNN	DETR
Best epoch	357	341	243
Precision (P)	0.963	0.969	0.932
Recall (R)	0.953	0.899	0.841
mAP@0.50	0.983	0.949	0.911
mAP@0.50:0.95	0.609	0.877	0.842

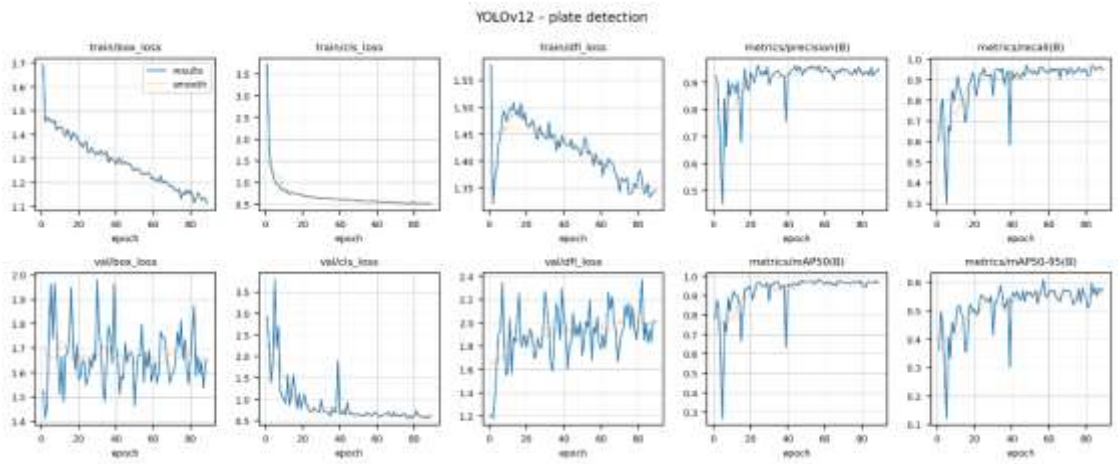


Figure 4.5 Training Metrics of Plate localization Model using YOLOv12

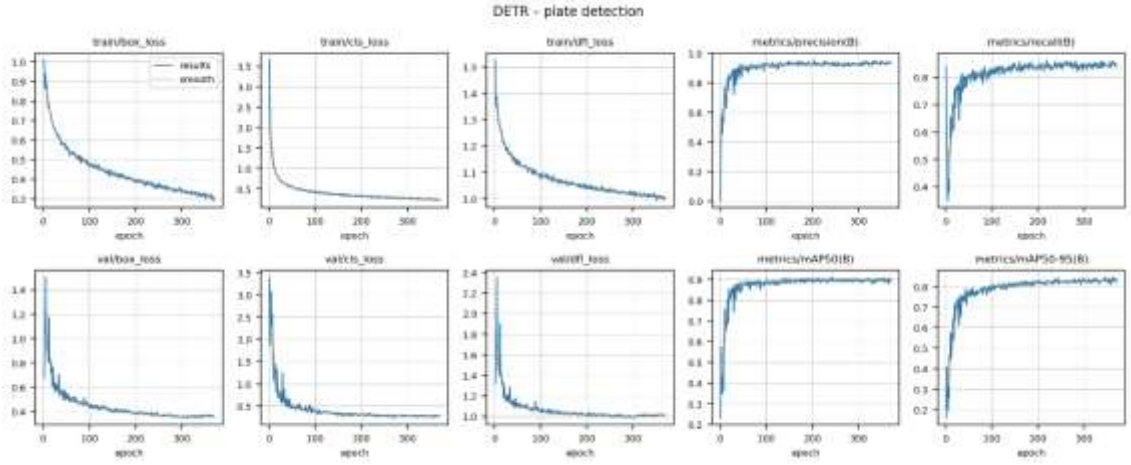


Figure 4.6 Training Metrics of Plate localization Model using Faster R-CNN

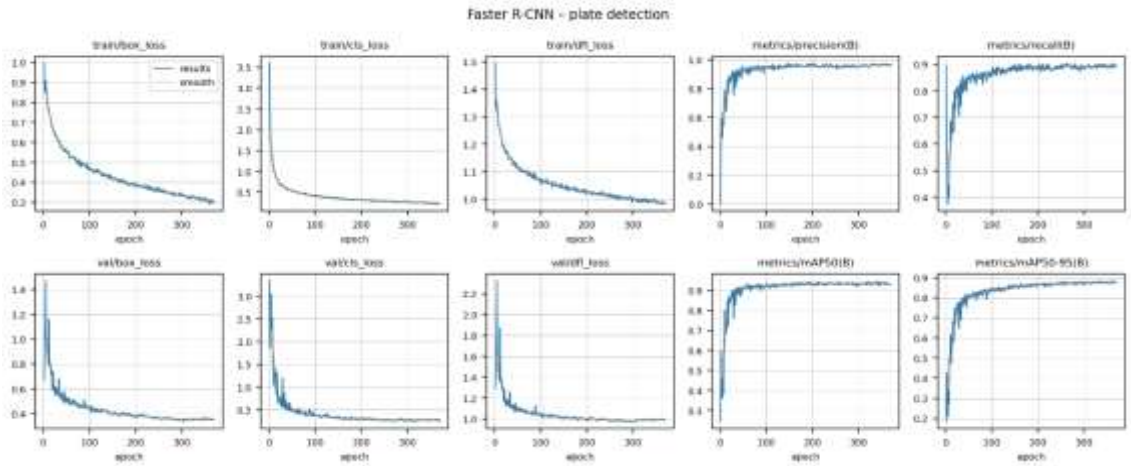


Figure 4.7 Training Metrics of Plate localization Model using DETR

The plate detector models performed so close to each. Also, the validation curves showing a similar tread which indicates better generalization rather than overfitting and almost every license plate in the vehicle is correctly predicted. Only a few background regions of plate are mistakenly predicted with a very small fraction of confusion. Overall, the YOLOv12 model is highly reliable to distinguish plate regions from non-plate region. Compared to heavy model of base paper, proposed light-weight model has achieved similar or slightly high accuracy with a convenient mAP score.

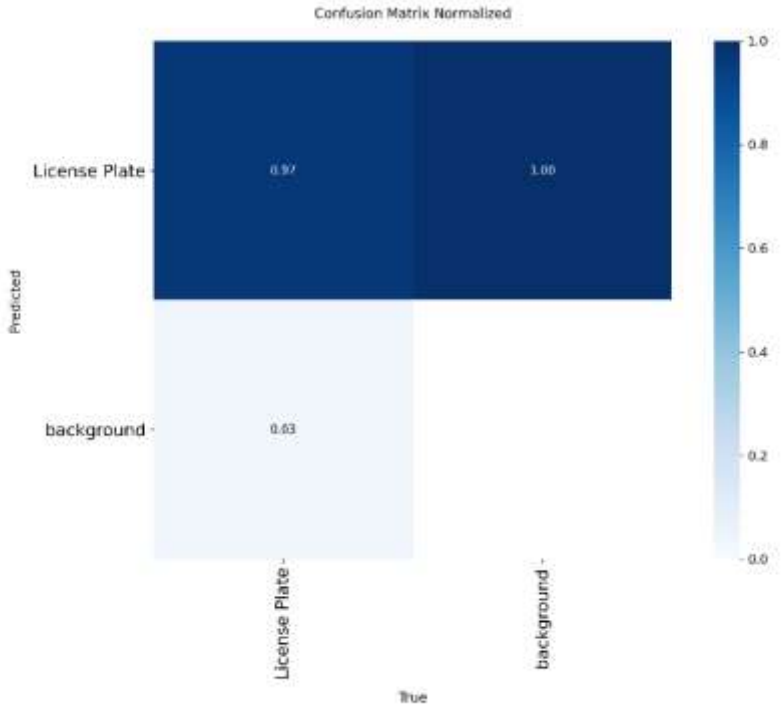


Figure 4.8 Confusion Metrics of Plate Localization Model using YOLOv12

4.1.3 Character Recognition Model

Table 4.3 Result Comparison Table of Character Recognition

Metric	YOLOv12	Faster R-CNN	DETR
Best epoch	121	112	131
Precision (P)	0.938	0.931	0.891
Recall (R)	0.933	0.875	0.834
mAP@0.50	0.967	0.924	0.890
mAP@0.50:0.95	0.684	0.642	0.597

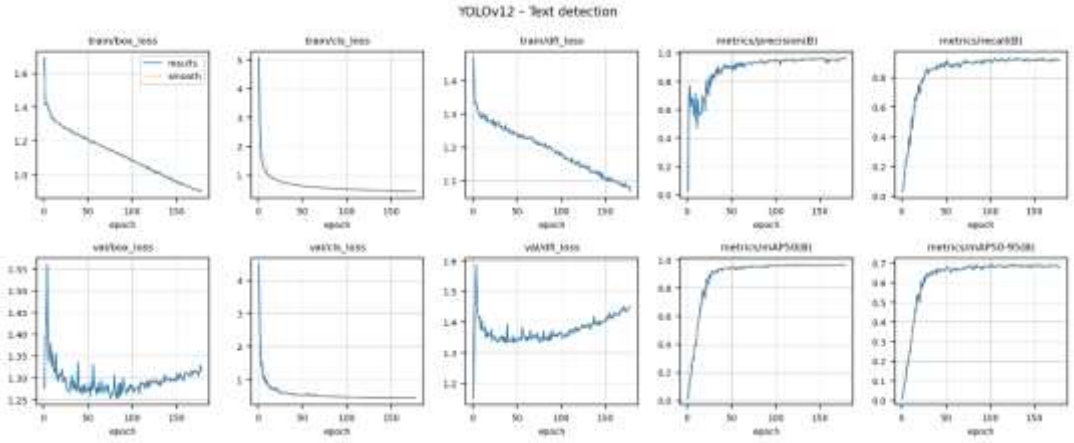


Figure 4.9 Training Metrics of Character Recognition Model using YOLOv12

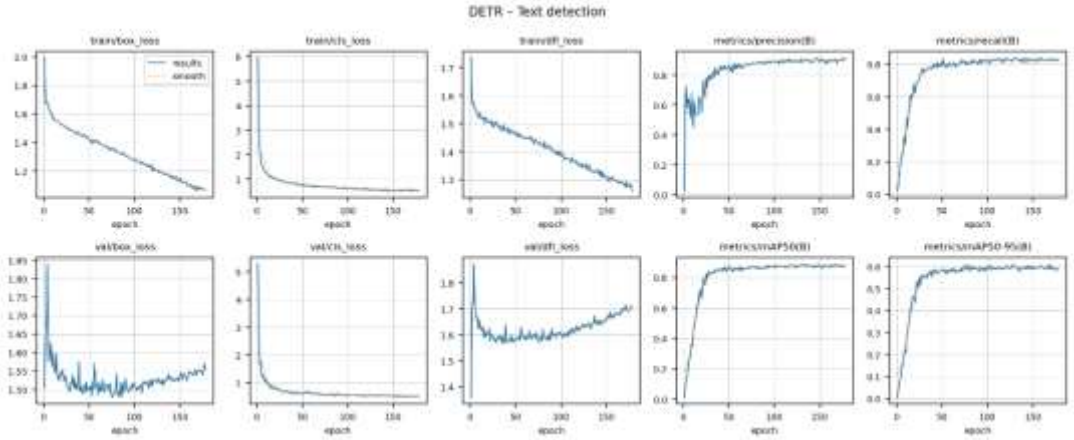


Figure 4.10 Training Metrics of Character Recognition Model using Faster R-CNN

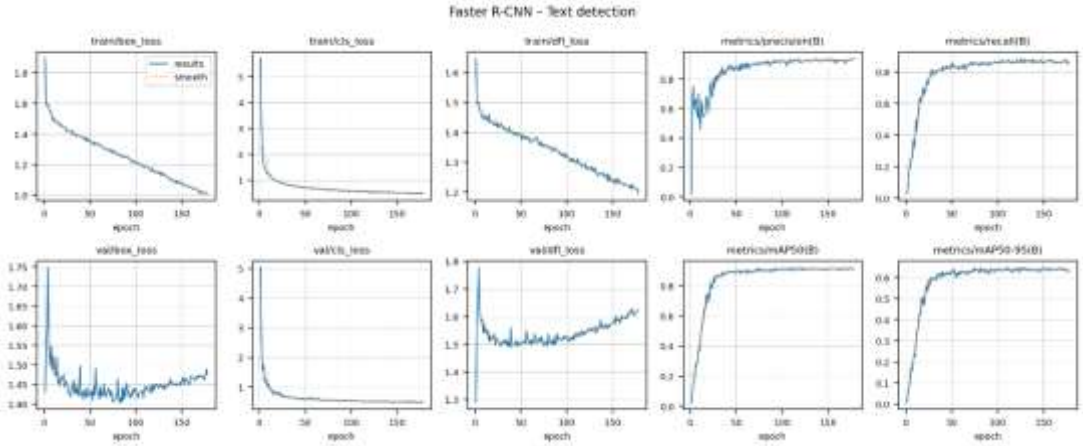


Figure 4.11 Training Metrics of Character Recognition Model using DETR

The YOLOv12 based Character recognition model shows a very reliable and stable training behaviour with the box, classification and DFL losses gradually reduced among all other models.. In first few epochs it improves the learning curve quickly and generalizes well to unseen plates. Though proposed model used the lightest version of YOLOv12, it still achieves almost same result as the heavy model like Faster R-CNN and DETR, in some case it performs slightly better which was the main goal of this study. There is a marginal drop in recall which indicates model misses some very small or degraded characters sometimes. At stricter IoU thresholds indicate the model localizes characters more precisely when the corresponding plate is clean cropped. Overall, YOLOv12 performs better in comparison to other models which is convenient to use in Bangla License plate recognition pipeline.

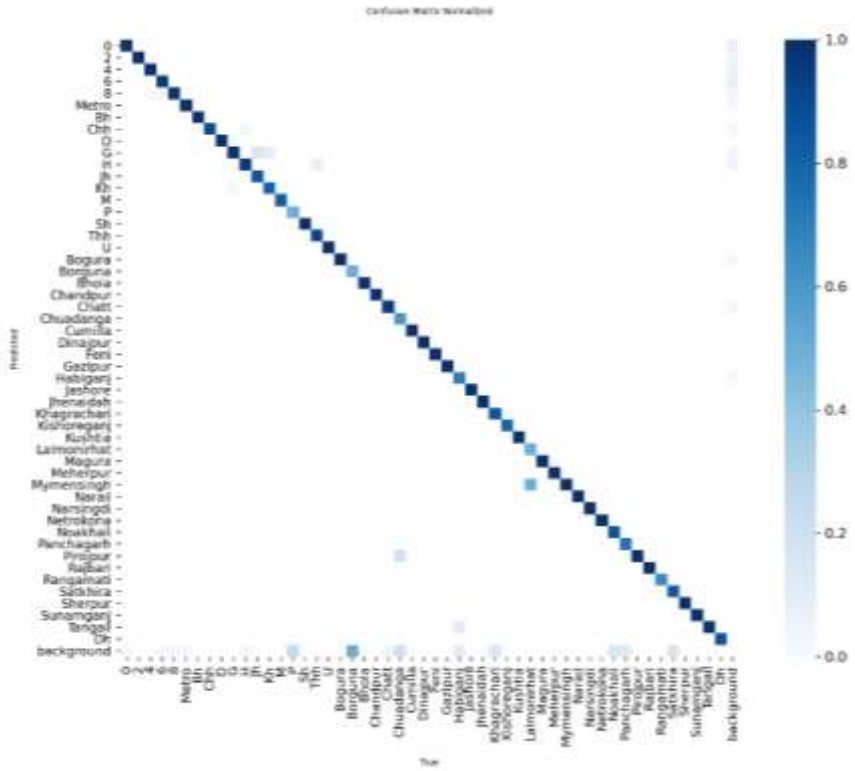


Figure 4.12 Confusion Metrics of Character Recognition

4.2 2 Layer Vs 3 Layer Result Comparison

This section represents a comparative study between two approaches to the License Plate Detection Pipeline. It compares how many licenses plates approach can detected from number of total plates present in the image and among them, how accurately models can recognize the characters on the plate.

4.2.1 License Plate Detection Result

Image	Total Number of Plates in Image	Detected Number of License Plates By 2 Layers Approach	Detected Number of License Plates By 3 Layers Approach
	2	1	2
	2	0	1
	3	2	2
	3	1	2
	4	2	4

To test the models in real world scenarios, some images with multiple vehicles are taken to see how well the models can predict unseen dataset. The plates inside the images are quite blurred and hard to see. In 2-layer approach, it messed up to detect all the plates present in the images due to background of the vehicles. 3-layer approach exactly made to address this problem, and it perform far well than 2 layer approach since it detects the vehicle first and then proceed with plate and characters.

4.2.2 Text Detection Result

Actual number in license plates	Detected number of license plates in 2 layers approach		Detected number of license plates in 3 layers approach	
	Detected number	Percentage of correctness	Detected Number	Percentage of correctness
ঢাকা-মেট্রো-ন ৯১-৯৫০২	ঢাকা-মেট্রো ১১-৯৫০২	80%	ঢাকা-মেট্রো-ন ৯১-৯৫০২	100%
ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	100%	ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	100%
ঢাকা-মেট্রো-ব ১৩-২৫৭০	ঢাকা-মেট্রো-ব ১৩-২৫৭০	100%	ঢাকা-মেট্রো-ব ১৩-২৫৭০	100%
ঢাকা-মেট্রো-ব ১২-৪০৩৯	Couldn't detect the plate	-	ঢাকা-মেট্রো-ব ১২-৪০৩৯	100%
ঢাকা-মেট্রো-স ১১-০২৭১	Couldn't detect the plate	-	ঢাকা-মেট্রো-চ ১১-০২৭১	93%
ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	100%	ঢাকা-মেট্রো-ব ১৫-৯৬৮৭	100%
ঢাকা-মেট্রো-ব ১২-৪০৩৮	Couldn't detect the plate	-	ঢাকা-মেট্রো-ব ১২-৪০৩৮	100%
ঢাকা-মেট্রো- ব ১১-৯৫০২	ঢাকা-মেট্রো-ম ১১-৯৫০২	93%	ঢাকা-মেট্রো- ন ১১-৯৫০২	93%

The result shows how perfectly 3-layer approach outperforms the 2-layer approach. In some cases, 2-layer approach couldn't detect the plate at all where 3-layer not only detect

the plates but recognized the characters as well. Since there is an extra layer to reduce the confusion with the background, now the plate detection model can do well and leads to a better performance with a perfect percentage to accurately detect Bangla license plate numbers.

4.3 Discussion

The overall result demonstrates the meaningful improvement and efficiency of the proposed multi-stage YOLOv12 pipeline for detecting Bangla License plate in real world scenes of traffic. Each of the model of the pipeline showed outperformance on not only test data, as well as additional real-world images with multiple vehicle existence.

Both vehicle detection and plate detection model showed strong and reliable result. The training curves and validation curves were smooth and stable from the beginning in the vehicle detection model which indicates model learned exact features and patters and did not overfit to the training data. As a result, the vehicle detection model precisely detects the clean and accurate region which improves the next stage performance by plate localization model. It helps to distinguish the background of the image and focus only the vehicle area to detect plate. It occasionally misses some very difficult plates which is fair enough because the model reduces inaccurate crops that could confuse the recognition stage. These benefits were carried out to the recognition stage to level of digits, Bangla letters and marks by the character recognition model.

The comparative study demonstrated value of the proposed architecture. 2-layer approach often fails to detect all plate when the background is cluttered, or the plates are rarely visible or blurred in multi-vehicle images. The 3-layer approach solves this problem by adding an extra layer to detect all the vehicle in the image first and then search for plates and character inside it. As a result, it detects more plates per image and gives higher accuracy in character recognition. This proves adding an extra helping layer at first can improve the result and increase the robustness of the whole pipeline.

CHAPTER 5

CONCLUSION

5.1 Findings & Contributions

The proposed workflow and architecture of this study built a end-to-end Bangla License plate recognition pipeline that can be used in real traffic areas of Bangladesh. The system used the latest YOLOv12 model for identifying vehicles in the frame, license plates and individual characters on that. The system can precisely identify 102 individual classes such as Bangla letters, district name, special markers of plate and area code with the lightest version of YOLOv12 which makes it practical for use in Bangladesh. This system is easier to deploy on normal computers without high configured hardware, and it can be easily maintained in situations where powerful GPUs or expensive servers are not available.

To reduce false positives of the system from background text regions, signboards or other plate-like structures in the image, this proposed system used an extra layer at the beginning which decreases the plate detector models' pressure by detecting all the vehicle present in the frame and crop their region. For this extra layer, plate detection model can now focus only in the vehicle area which increases the accuracy of the model compared to two-layer model, especially when the images contain multiple vehicles. The more it detects the plate correctly, the better it can detect the characters on it. So, this three-layer pipeline: "vehicle detection → plate detection → character detection" improves the overall performance and gives better coverage than the traditional two-layer pipeline of "plate detection → character detection"

Additionally, this system aims to be evaluated with real traffic scene where the frame consists of multiple types of vehicles. The system is trained on the dataset which contain images with only a single vehicle and clearly visible license plate—which isn't usual scenario. So for evaluation we use a set of test images with multiple vehicle in the frame with various location, situation and conditions. Both the two-layer and three-layer

model pipelines are used to test on these images. The result highlights both the novelty and the practical usefulness of proposed three-layer approach which only works well on the clean dataset but also handles challenging multi-vehicle scenes much more effectively.

5.2 Limitations

Although the models perform very well even with the lightweight version of YOLOv12, some limitations still remain. Small, blurry, or heavily occluded plates in the images are sometimes missed. In addition, the model can be confused by visually similar district names. For example, গোপালগঞ্জ and সিরাজগঞ্জ share the same ending (গঞ্জ), so the model may occasionally replace one district with the other based on visual similarity rather than the full word.

5.3 Recommendations for Future Works

Based on the current result and limitations, there are some recommended ways to improve this system more. One of the important future scopes is to collect vehicle images in large scale with complex conditions such as small, blurry, night view, heavily occluded plates. The images should contain multiple vehicles so that models can be trained with the images containing more than one plate in the frame. Instead of single images, testing the models on continuous video streams can make the system more practical in real deployments.

To overcome the error for visually similar spelling of district name, the system could use a list common plate patterns of valid district names in a large amount of dataset. The system can join the predicted characters into a string and compare this string with the closest match district name using edit distance or string similarity.

CHAPTER 6

REFERENCES

1. Hussain, M. (2024, July). YOLOv5, YOLOv8 and YOLOv10: The go-to detectors for real-time vision. arXiv. https://doi.org/10.48550/arXiv.2407.02988

2. Khanam, R., & Hussain, M. (2025, February 20). A review of YOLOv12: Attention-based enhancements vs. previous versions. arXiv:2502.14740 [cs.CV].

3. Tian, Y., Ye, Q., & Doermann, D. (2025, February 18). YOLOv12: Attention-centric real-time object detectors. arXiv:2502.12524 [cs.CV].

4. Kundrotas, M., Janutėnaitė-Bogdanienė, J., & Šešok, D. (2023). Two-step algorithm for license plate identification using deep neural networks. *Applied Sciences*, 13(8), 4902. https://doi.org/10.3390/app13084902

5. Ismail, G. M., & Ahamed, I. (2025). YOLOv8-based license plate recognition for Bangladeshi vehicles. Authorea preprint. https://doi.org/10.22541/au.173748276.69471512/v1

6. Hossain, S., Kabir, M. A., & Islam, N. (2021). Challenges in Bangla license plate recognition: Script complexity and environmental factors. In Proceedings of IEEE ICCIT (pp. 1–6). IEEE.

7. Saif, N., Ahmmed, N., Pasha, S., Shahrin, M. S. K., Hasan, M., Islam, S., & Jameel, A. S. M. M. (2019). Automatic license plate recognition system for Bangla license plates using convolutional neural network. In IEEE Region 10 Conference (TENCON) (pp. 1–6). IEEE.

8. Shomee, H. H., & Sams, A. (2021). License plate detection and recognition system for all types of Bangladeshi vehicles using a multi-step deep learning model. In International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 1–8). IEEE.

9. Onim, M. S. H., Nyeem, H., Roy, K., Hasan, M., Ishmam, A., Akif, M. A. H., & Ovi, T. B. (2022). BLPnet: A new DNN model and Bengali OCR engine for Automatic License Plate Recognition. Journal of King Saud University – Computer and Information Sciences (preprint on arXiv:2202.12250).

10. Ashrafee, A., Khan, A. M., Irbaz, M. S., & Nasim, M. A. (2022). Real-time Bangla license plate recognition system for low resource video-based applications. In WACV 2022 Workshops – Real-World Surveillance (pp. 1–10).

11. Afrin, N., Hasan, M. M., Safin, M. F. E., Amin, K. R., Haque, M. Z., Ahmed, F., & Shawon, M. T. R. (2023). Bengali license plate recognition: Unveiling clarity with CNN and GFP-GAN. arXiv:2312.10701.

12. Mahmood, Z., Khan, K., Khan, U., Adil, S. H., Ali, S. S. A., & Shahzad, M. (2022). Towards automatic license plate detection. Sensors, 22(3), 1245. https://doi.org/10.3390/s22031245

13. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. arXiv:1506.02640.

14. Tusar, M. H., Bhuiya, M. T., Hossain, M. S., Tabassum, A., & Khan, R. (2022). Real-time Bangla license plate recognition with deep learning techniques. In International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) (pp. 1–6). IEEE.

15. Nasim, H. I., Printia, F. J., Himel, M. H., Rashid, R., Chowdhury, I. J., Mondal, J. J., & Hossain, M. S. (2024). Fog-resilient Bangla car plate recognition using Dark Channel Prior and YOLO. In WACV 2024 Workshops – WVLL (pp. 1–10).

16. Nayeem, M. J., & Mondal, M. N. I. (2025). An efficient approach to recognize Bangla license plate for diverse-quality images. In Human-Centric Smart Computing (ICHCSC 2024), Smart Innovation, Systems and Technologies (Vol. 440, pp. 147–162). Springer. https://doi.org/10.1007/978-981-96-3420-0_13

17. Saha, U., Ahamed, I. U., & Hossain, M. I. (2024). YOLOv8 for Bangla license plate recognition: Advancing real-time object detection in localized contexts. In IEEE International Conference on Informatics and Computational Sciences (ICICoS) (pp. 1–6). IEEE. https://doi.org/10.1109/ICICoS62600.2024.10636876

18. Moussaoui, H., ElAkkad, N., Benslimane, M., El-Shafai, W., Baihan, A., Hewage, C., & Rathore, R. S. (2024). Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition. Scientific Reports, 14, 14389. https://doi.org/10.1038/s41598-024-65272-1

19. Ahamed, I., & Ismail, G. M. (2025). YOLOv8-based license plate recognition for Bangladeshi vehicles. Authorea. https://doi.org/10.22541/au.173748276.69471512/v1

20. Ahamed, I., & Feng, W. (2024). Enhancing Bangladeshi license plate recognition: A YOLOv8 approach with Roboflow integration for accuracy and speed optimization. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 12(5). https://doi.org/10.22214/ijraset.2024.61520

21. Bappy, H., & Talukder, K. H. (2024). Real-time vehicle license plate recognition (VLPR) using deep CNN. International Journal of Fundamental and Multidisciplinary Research (IJFMR), 6(3), 1–10.

22. Das, A., & Hasan, K. M. A. (2024). YOLOv9-powered Bangla license plate recognition: A comparative analysis for optimized performance in localized contexts. (Preprint / conference paper).

23. Barka, S., Manongga, D., Hendry, & Aminuddin, A. (2023). Enhanced YOLOv8-based system for automatic number plate recognition. Technologies, 12(9), 164. https://doi.org/10.3390/technologies12090164

24. Satya, B., Manongga, D., Hendry, & Aminuddin, A. (2025). Optimized YOLOv8 for automatic license plate recognition on resource-constrained devices. Engineering, Technology & Applied Science Research, 15(2), 21976–21981. https://doi.org/10.48084/etasr.9983

25. Sabu, [Initials]. (2025). Automated license plate recognition using YOLOv8 and EasyOCR. In Congress on Intelligent Systems, Lecture Notes in Networks and Systems (Vol. 1278, pp. 1–10). Springer.

26. Chen, L., Li, W., & Zhang, Q. (2023). Attention-enhanced YOLOv8 for Chinese license plate recognition. In Proceedings of the IEEE Intelligent Vehicles Symposium (pp. 1–6). IEEE.

27. Das, A., Ghosh, S. K., & Saha, P. K. (2022). CRNN-based Bangla character recognition from license plates. Journal of Imaging, 8(5), 132.

28. Fahim, S. H., Rasel, A. A. S., Sarker, A. R., & Chowdhury, T. (2025). Bangla License Plate Detection Using YOLO v8 Model. *International Journal of Engineering and Computer Science*, 12(1), 15–26. Retrieved from https://probejournals.com/wp-content/uploads/2025/06/jb-12-1.pdf

29. Zheng, Z., Wang, P., & Liu, W. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 12993–13000.

30. Wang, Y., Zhang, X., & Yang, L. (2023). Focal-EIoU: A robust loss function for object detection. In Proceedings of IEEE ICIP (pp. 438–442). IEEE.

31. Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2024). YOLOv12: Enhanced object detection with attention-guided feature fusion. arXiv:2405.14458.

32. Srivastava, B., & Chahal, P. (2025). Real-time anomaly reduction in license plate recognition using YOLO with attention mechanism and data augmentation. International Journal of Information Technology. https://doi.org/10.1007/s41870-025-02810-8

33. Zhu, R., He, Q., Jin, H., Han, Y., & Jiang, K. (2025). License plate detection based on improved YOLOv8n network. Electronics, 14(10), 2065. https://doi.org/10.3390/electronics14102065

34. Nguyen, T. (2025). Improve underwater object detection through YOLOv12 architecture and physics-informed augmentation. arXiv preprint. https://doi.org/10.48550/arXiv.2506.23505

35. Shomee, H. H., & Sams, A. (2021, April 25). Bangla LPDB - A (Version v1) [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.4718238

PLAGIARISM REPORT

221-35-1009

ORIGINALITY REPORT

15%	12%	12%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Human-Centric Smart Computing", Springer Science and Business Media LLC, 2025 Publication	1%
2	www.mdpi.com Internet Source	1%
3	arxiv.org Internet Source	1%
4	www.ijraset.com Internet Source	1%
5	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
6	Bhawana Srivastava, Poonam Chahal. "Real-time anomaly reduction in license plate recognition using YOLO with attention mechanism and data augmentation", International Journal of Information Technology, 2025 Publication	<1%

ACCOUNT CLEARANCE

LIBRARY CLEARANCE