

Dark Channel Prior (DCP) Based Bangla Car Plate Detection and Recognition in Foggy Weather

by

Hamim Ibne Nasim

20301443

Fateha Jannat Printia

20301357

Mahamudul Hasan Himel

20301397

Rubaba Rashid

20301267

Iffat Jahan Chowdhury

20301050

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

© 2024. Brac University
All rights reserved.

Declaration

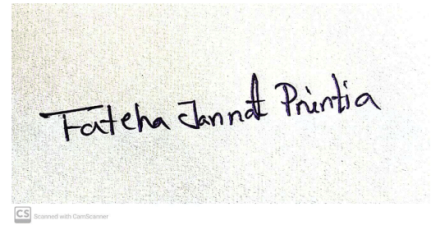
It is hereby declared that

1. The provided thesis is our own unique work for our degree at Brac University.
2. The thesis contains no previously published or authored by a third party content, unless properly referenced via comprehensive and correct referencing.
3. The thesis contains no material that has been approved or submitted for consideration for any other degree or certificate at a university or other institution.
4. We have acknowledged every major source of assistance.

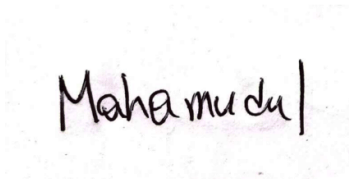
Student's Full Name & Signature:



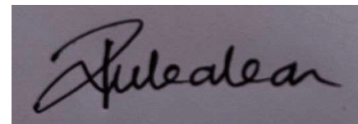
Hamim Ibne Nasim
20301443



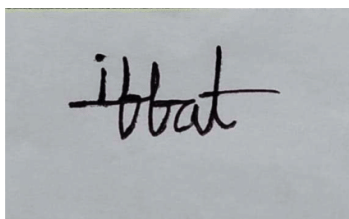
Fateha Jannat Printia
20301357



Mahamudul Hasan Himel
20301397



Rubaba Rashid
20301267



Iffat Jahan Chowdhury
20301050

Approval

The thesis titled “Dark Channel Prior (DCP) Based Bangla Car Plate Detection and Recognition in Foggy Weather” submitted by

1. Hamim Ibne Nasim (20301443)
2. Fateha Jannat Printia (20301357)
3. Mahamudul Hasan Himel (20301397)
4. Rubaba Rashid (20301267)
5. Iffat Jahan Chowdhury (20301050)

of Fall, 2023 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in Computer Science on January 9, 2024.

Examining Committee:

Supervisor:
(Member) Supervisor:



Jannatun Noor

Assistant Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Head of Department:
(Chairperson)

Sadia Hamid Kazi, Ph.D.

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

For the avoidance of doubt, the aforementioned study work has not infringed upon any human rights, welfare, or dignity provisions. In addition to cited references from a variety of credible sources, the study incorporates the truthfulness and integrity of the research participants. Neither the data collecting process nor the outcomes have been influenced by any bias or prejudice. We think our effort contributes to the long-term well-being of mankind and demonstrates our respect for all intellectual property.

Accepted Research Works from This Thesis

Research Article,

- Workshop on Vision-Based Understanding for Low-Resource Languages (WVLL), WACV Workshop 2023: Nasim, H. I., Printia, F. J., Hasan, M., Rashid, R., Chowdhury, I. J., Mondal, J. J., Islam, M.F. & Noor, J. (2024). Fog-Resilient Bangla Car Plate Recognition Using Dark Channel Prior and YOLO. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1110-1119).

Poster,

- 2023 10th International Conference on Networking, Systems and Security: Nasim, H. I., Printia, F. J., Rashid, R., Himel, M. H., Chowdhury, I. J., Islam, M. F., & Noor, J. Fog-Resilient Bangla Car Plate Recognition using Dark Channel Prior and YOLO.

Abstract

The intention of autonomous license plate (LP) detection is to find an LP spot in a picture without any kind of human intervention. Automatic license plate detection (ALPD) techniques have been developed; however, the majority of them do not take into account the numerous potentially hazardous image cases that frequently arise during actual driving scenarios. Hazardous picture conditions include low contrast settings, objects that are comparable to LP in the background, LP areas that are inclined horizontally, and weather effects like rain or fog. In this study, we present a novel approach to localizing and recognizing Bangla car plates in foggy or rainy weather using the Dark Channel Prior (DCP) approach for dehazing foggy images. There are three parts to the suggested method. In the first stage, the DCP dehazing algorithm is used to reduce the foggy effect on the input images. In the second stage, a YOLOv8 object detection model is used to detect the Bangla license plates from the dehazed images and lastly, the OCR technique is used to recognize and extract texts from the identified images of the license plates. Thus, our study aims to use DCP, the YOLOv8 algorithm, and the OCR technique to identify and recognize Bangla vehicle plates in foggy conditions in order to improve transportation safety, law enforcement, traffic flow, and tax revenue collection.

Keywords: YOLOv8, Automatic License Plate Detection, Image Processing, Dark Channel Prior, EasyOCR, Neural Networks, Deep Learning

Acknowledgements

We are grateful to Allah, the Almighty, for providing us with the resources and the will to complete our studies on time. We express our gratitude to the anonymous reviewers who attended the WVLL Workshop from WACV 2023 and provided us with valuable feedback and direction on how to rectify the issues identified. With that being stated, we wish to express our appreciation to our external collaborator Joyanta Jyoti Mondal from the University of Alabama at Birmingham, and Md. Farhadul Islam, Research Assistant at Brac University, and Dr. Jannatun Noor, our supervisor, for providing invaluable guidance throughout the duration of the study. Last but not least, we are grateful to our loving parents, who have never left our side and who constantly pray for us.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iv
Accepted Research Works from This Thesis	v
Abstract	vi
Acknowledgment	vii
Table of Contents	viii
List of Figures	x
List of Tables	xii
Nomenclature	xiv
1 Introduction	1
1.1 Motivation	2
1.2 Problem Statement	2
1.3 Limitation of Existing Literature	3
1.4 Research Objectives	4
1.5 Contribution	5
1.6 Structure of Our Thesis Book	6
2 Literature Review	7
2.1 Basic Object Detection Models	7
2.2 Advanced Dehazing Methods	10
3 Background	14
3.1 Dehazing with Dark Channel Prior Method	14
3.1.1 Atmospheric Light Analysis	15
3.1.2 Transmission Map Forecast	15
3.1.3 Transmission Refinement via Guided Filtration	15
3.1.4 Image Restoration	15
3.2 Detecting Object Using YOLOv8	16
3.3 Recognition Using OCR	18

3.3.1	Comparison Between EasyOCR and Other Alternative Techniques for Text Recognition from Images	19
4	Dataset	20
4.1	Dataset Analysis	20
4.2	Customised Dataset	21
5	Research Methodology	25
5.1	Workflow	25
5.2	Data Pre-Processing Using DCP	26
5.3	Proposed Model	28
5.4	Detection Using YOLOv8	30
5.5	Recognition Using OCR	35
5.5.1	Setting Up OCR Reader	35
5.5.2	Object Detection with YOLOv8	36
5.5.3	Image Cropping (If Objects Detected)	36
5.5.4	OCR on Cropped Images	37
5.5.5	Processing OCR Results	37
5.5.6	Calculating Average Confidence Score	37
6	Experimental Evaluation	39
6.1	Experimental Setup	39
6.1.1	Basic libraries for Our Model	39
6.1.2	Basic libraries for OCR	40
6.2	Experimental Findings	40
6.2.1	Customised Dataset's Findings	40
6.2.2	Qualitative Evaluation	41
6.2.3	Quantitative Evaluation	43
6.2.4	Findings from YOLOv8	48
6.2.5	Before Applying DCP and After Applying DCP	49
6.2.6	Performance Evaluation of EasyOCR	50
6.2.7	Overview of Findings	50
7	Limitations and Future Work	51
7.1	Limitations	51
7.2	Future Work	53
8	Discussion	54
9	Conclusion	56
	Bibliography	61

List of Figures

1.1	Representation of Vehicle Number Plate in Bangladesh [37].	1
3.1	Dark Channel Prior Architecture	14
3.2	Model Structure of YOLOv8 [64]	16
3.3	The Structure of YOLOv8 With Backbone	17
3.4	OCR Workflow	18
4.1	Dataset Split and Class Variety Visualization	21
4.2	Approach Combines a Depth Network, Pose Network, Per-Pixel Minimum Reprojection Loss, and Full-Resolution Multi-Scale Processing for Enhanced Depth Estimation [32].	22
4.3	Mean Intensity of Ground Truth Images and Customized Foggy Images	23
4.4	Variation in Density Mean Intensities of the Individual Image	24
5.1	Work Flow	25
5.2	Dark Channel Image and Transmission Map Forecast Image	26
5.3	Transmission Refinement via Guided Filtration and Image Restoration	26
5.4	Pre-processing Using DCP	27
5.5	Proposed Model Architecture	28
5.6	Proposed Model Working Procedure	30
5.7	YOLOv8 Workflow	30
5.8	Training and Validation Metrics of YOLOv8 Model	32
5.9	Training and Validation Loss of YOLOv8 Model	33
5.10	Confidence Curves for YOLOv8	34
5.11	Test Result Prediction of Trained Model	34
5.12	Precision-Recall Curve	35
5.13	Easy-OCR Workflow	35
5.14	Object Detection With YOLOv8	36
5.15	Cropped Image	37
5.16	OCR on Cropped Image	37
5.17	OCR Recognition: True vs. False	38
6.1	Visual Comparison of Ground Truth, Real Foggy Images and Monoculardepth Foggy Images	40
6.2	Frequency Intensity of Ground Truth, Real Foggy Images and Monoculardepth Foggy Images	41

6.3	Performing a Qualitative Analysis of 10 Images From Our Customize Unique Dataset Using Several Algorithms (a) Hazy Images With Fog, (b) He et al.[3], (c) Berman et al.[14], (d) Xie et al. Method, (e) Kumari et al. Method, (f) Meng et al. Method, and (g) Cai et al. Method, (h) Proposed Method.	42
6.4	Comparative Visualization of Transmission Maps for Three Sample Images Processed by Dehazing Algorithms	43
6.5	Comparative Analysis of Seven Image Processing Algorithms on Three Images From a Custom Dataset, Showcasing the Original Alongside Each Algorithm's Output.	43
6.6	SSIM Graph for Dataset-1	44
6.7	SSIM Graph for Dataset-2	44
6.8	PSNR Graph for Dataset-1	46
6.9	PSNR Graph for Dataset-2	46
6.10	Before applying DCP and after applying DCP	49
6.11	Wrong Sequence LP	50
7.1	Difficult to Identify Characters [37]	52

List of Tables

4.1	Comparison of Vehicle Types in Dataset-1 and Dataset-2	20
5.1	EasyOCR Average Confidence Score in Different Models for Both Datasets	38
6.1	SSIM Values of Different Algorithm on Dataset-1	45
6.2	SSIM Values of Different Algorithm on Dataset-2	45
6.3	PSNR Values of Different Algorithm on Dataset-1	47
6.4	PSNR Values of Different Algorithm on Dataset-2	47
6.5	NIQE Values of Different Algorithm on Dataset-1	48
6.6	Accuracy Evaluation of YOLO Models Across Two Datasets	49
6.7	Parameters Overview	50

Nomenclature

The following list describes several symbols and abbreviations that will be later used within the document's body.

ROI Return on Investment

SSIM Structural Image Similarity

ALPD Automatic license plate detection

DCP Dark Channel Prior

LP License Plate

NIQE Naturalness Image Quality Evaluator

OCR Optical Character Recognition

YOLOv8 You Only Look Once Version 8

ALPR Automatic License Plate Recognition

API Application Programming Interface

BRTA Bangladesh Road Transport Authority

CCPD Chinese City Parking Dataset

CCTV Closed Circuit Television

CLAHE Contrast-Limited Adaptive Histogram Equalization

CNG Compressed Natural Gas

CNN Convolutional Neural Network

CPU Central Processing Unit

CTC Connectionist Temporal Classification

CVAT Computer Vision Annotation Tool

DCM Dark Channel Map

FADE Fast Atomic Density Evaluation

FC Fully Connected

FPS Frame Per Second
GPU Graphics Processing Unit
GRU Gated Recurrent Units
HOG Histogram of Oriented Gradients
KITTI Karlsruhe Institute of Technology and Toyota Technological Institute
LPR LP Recognition
LSE Language Sensitive Editor
MSCNN Multi-Scale Convolutional Neural Network
MSE Mean Square Evaluation
PSNR Peak Signal-to-Noise Ratio
SSD Solid State Drive
SVM Support Vector Machine

Chapter 1

Introduction

As a result of excessive development and the quick expansion of car companies, automobiles have developed into an essential component of our lives. According to the BRTA, 3.6 million cars have been registered in Bangladesh in 2023 [54]. However, the parking lot infrastructure needs to be improved, and keeping track of automobiles as they arrive and exit is complicated and disorganized. Moreover, finding a parking space in a city may be challenging because of the number of automobiles on the road. In Bangladesh, where data quality is an issue, it might be tough to determine whether cars have been involved in any illegal activity. Additionally, not all LP layouts are the same, and as the lighting condition changes frequently, dealing with this situation is another difficulty. That is where additional work must be done on the method to get a reliable outcome. So, keeping track of them is essential for various reasons, including transportation, the environment, public safety, law enforcement (in the case of unregistered vehicles or illegal activities), traffic control, revenue collection, and toll purposes. However, it might be challenging to monitor automobiles when dangerous circumstances occur, such as when it is raining or hazy outside. Figure 1.1 displays several Bangla LPs in the style the Bangladesh Road Transport Authority (BRTA) requires.

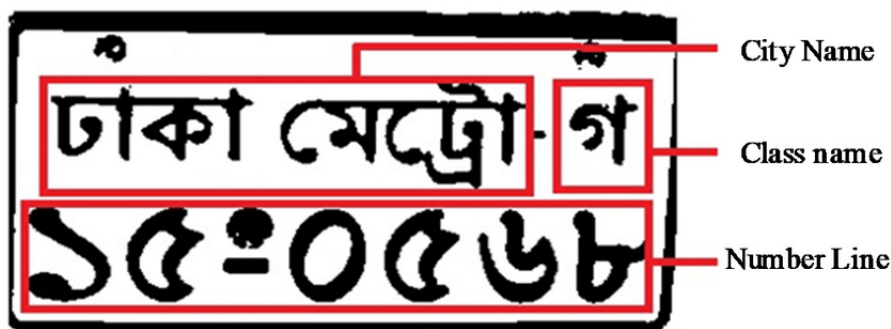


Figure 1.1: Representation of Vehicle Number Plate in Bangladesh [37].

1.1 Motivation

Keeping track of all the automobiles in a country is essential to ensure public safety, running an efficient traffic control operation, and collecting toll money. However, the challenge of locating or identifying a vehicle based on its LP is a considerable one, mainly when the weather is unfavorable or when there are physical factors such as corrosion, desorption, or fluctuations in the size and font of the plate. Consequently, there is an urge for trustworthy technology that can reliably spot and identify the LP of cars in surroundings with low levels of light and fog.

Potential applications in numerous fields are listed below, all of which suggest that the work being offered will be helpful in the real world.

- To do law enforcement in a safe way, we need to keep our transportation system in good shape so we can keep an eye on tolls and unregistered vehicles. But it is hard to watch them when it is hazy outside. So, our goal is to ensure law enforcement can trust dangerous situations.
- Regarding traffic regulation, parking is one of the most pressing issues in Bangladesh's leading cities. Taking pictures of LPs and sending them to an automated system is a convenient way for city authorities to track out a car's rightful owner or driver when it has been improperly parked. The system may provide helpful information for managing this situation by responding with the name and phone number of the owner or driver of the car. Conversely, traffic flows more smoothly when entrances and exits are well-managed.
- This technology combines LP readers with road detectors to keep drivers and pedestrians safe on the road. This way, cars can avoid risky situations like driving through hazardous weather.
- A fully ALPR system could help to build a good system for giving out authorizations, collecting taxes, and managing traffic. This part of a digital government will be helpful.

Since an automobile's fundamental information can be gathered from its LP, we have focused on that.

1.2 Problem Statement

Our study, which goes by the title "Dark Channel Prior (DCP) Based Bangla Car Plate Detection and Recognition in Foggy Weather," tackles the challenge of detecting and identifying the number plates of cars in circumstances where visibility is low, such as in fog. Occlusion, reflection, and physical characteristics of the vehicles, such as corrosion or desorption, are some of the challenges that must be overcome. Moreover, many Bangladeshis can now afford their cars because of the country's improving economy. There must be more LPs when more cars are on the road. There are a variety of challenges in determining a vehicle's identification from its LP in Bangladesh.

To begin, every single LP displays a number in the Bangla script. Unlike in other

countries, our automobile registration numbers are printed in many media. For instance, a person who buys a motorbike today must register it before legally riding it on public roads. His machine-engraved LP may take up to a year to arrive. In this situation, they laminate the printed number and affix it to the vehicle’s service records. The number appears different in a machine-engraved number plate since there is no font for writing numerals on paper. This kind of printed number requires its own massive dataset.

Additionally, unlike first-world countries, we do not have high-resolution traffic cameras. Another significant challenge is using low-resolution closed-circuit video to read LPs from moving objects (cars, buses, and motorcycles). Noisy pictures are difficult to recognize automatically in low-resolution, noisy video. In addition to that, the perspectives captured by the camera also play a significant part in the identification process.

1.3 Limitation of Existing Literature

This study practicality depends on resolving many possible challenges, each of which presents unique obstacles to effectively implement the suggested fog-resistant automobile plate recognition system. There are numerous factors that might prevent the study from being conducted. These are some of them:

- **Inadequate noise and excessive haze removal:** We use the DCP algorithm to remove the haze from the pictures. While it is known for its effectiveness, sometimes the DCP algorithm may not be able to remove all of the noise in the photographs, which may result in an inaccurate reading of the LP. Moreover, DCP may sometimes demonstrate an excessively aggressive elimination of haze, leading to the loss of detailed components and textures in the dehazed pictures. This may lead to unrealistic or artificial outputs, particularly in complex scenarios with diverse atmospheric conditions. And it can also be a reason for an inaccurate result. Therefore, the DCP algorithm’s drawbacks, such as inadequate noise reduction and aggressive haze elimination, are possible restraints in the context of this research.
- **Occlusion:** When any object completely blocks the LP out, the system may face significant difficulty identifying the LP. When we built our dataset, we found that many LPs had other objects or obstructions attached. The presence of foreign materials or impediments, such as dirt, stickers, BRTA seal, or other components on the LP, may hamper the correct recognition process of the Bangla car plates. Moreover, these obstacles can affect the visual clarity of the characters, which can result in possible mistakes in identification. Specifically, these obstacles may affect the functionality of the OCR system, which entirely depends on the clear and distinct images of characters for accurate interpretation. Foreign items attached to the number plate may cover sections of the letters, change the way they appear, or introduce additional visual noise. So this can make it difficult for the system to accurately extract and identify the required information from the number plate.
- **Limited Data:** The performance of deep learning algorithms such as YOLOv8 strongly depends on the quantity and quality of the training data. If it is foggy

or dark outside, the system may not be as accurate since it cannot read as much information on the LPs. We have created foggy images with Monocular depth estimation, which performs like real fog. Nevertheless, it is crucial to acknowledge that the complexities and intricacies of real fog may be partially captured in artificially generated images. Therefore, the lack of authentic foggy images can limit real-world performance with unpredictable atmospheric conditions.

- **Computational Needs:** The proposed approach, especially the DCP algorithm, may have many computational requirements. The execution of DCP may need substantial computational capacity, especially when dealing with large datasets in real-time applications. The computational load increases when a large volume of data has to be processed at a time, particularly where real-time analysis is crucial, for example, in dynamic traffic environments. Therefore, the high computing requirements of the DCP algorithm may provide obstacles that could delay the seamless integration of the method and may raise concerns over its practicality in real-world scenarios, where limited resources and the need for quick answers are crucial.
- **Inadequate LP Outlines:** Many LPs need proper formatting. LPs that lack outlines are relatively easy to categorize and identify. Moreover, when the color of the vehicle completely matches the color of the LP, it becomes challenging to distinguish them and identify the LP separately. That is why YOLO and OCR’s workflow can also get hampered. The presence of similar color tones might result in visual ambiguities, which may lead to misinterpretations and errors during the identification process. So, it can be another potential limitation for this particular research.
- **Unavailability of Foggy Bengali Car Plate Dataset:** Since Bangla is considered to be a language with limited resources, there is currently no dataset available for the language that contains LPs for cars that are affected by fog or hazardous conditions. Because we are attempting to address issues in potentially hazardous environments, it is difficult to develop specific criteria. In considering this, we require photographs that were blurry because, in potentially hazardous circumstances, LPs are not readable. So far, we have not been successful in locating a dataset that is under the requirements that we have specified.

1.4 Research Objectives

The purpose of the study, which is referred to as “Dark Channel Prior (DCP)-Based Bangla Car Plate Detection and Recognition in Foggy Weather”, is to find a method that is effective in an image that has a lower intensity in the local patch of the majority of the pixels that make up that image. To improve the clarity of the photographs, an algorithm known as DCP reduces the amount of haze in the atmosphere. A portion of the light is said to have been transmitted when it travels through the atmosphere and reaches the camera. Once it arrives at the camera, it is influenced by the density of the haze as well as the distance between the camera and the vehicle. YOLOv8, a deep neural network design known as CNN, is used to discover and identify the Bangla number plate from the dehazed images. Finally,

a Python-based optical character recognition technique, EasyOCR, is used to recognize the texts from the identified LPs. The DCP model reduces the haze in an image by utilizing the relationship between haze density and transmission. The recommended approach uses all three of these methods simultaneously. The ultimate goal of this study is to improve and enhance the efficiency of Bangladesh’s transportation system. This will be accomplished by simplifying the process of keeping track of cars, ensuring that they are correctly listed, paying tolls, also locating and following vehicles being used for illegal activities.

1.5 Contribution

Some of the most important contributions that this study has made are as follows:

- This study presents a custom dataset in the Bengali language that specifically focuses on foggy environments. It aims to fill the gap in the availability of comprehensive Bangla datasets for training and assessing models for recognizing automobile LPs in foggy environments. This dataset is a valuable resource for researchers and practitioners currently working in this field. In Bangladesh, our study discovered many recent studies that focused on automatic car plate recognition systems in the Bangla language. Having said that, most of these studies were conducted in clear weather [34]. In contrast, many automatic car plate recognition systems optimized for foggy conditions were found, but none of them were explicitly tailored to the Bangla language [42]. So introducing this unique dataset is a valuable contribution in the field of Computer Vision.
- Our study presents an innovative method for reducing haze from Bangla car plate images by using the improved DCP algorithm. This method enhances visibility and increases the quality of images shot under hazy conditions. Compared to previous algorithms, our suggested model performed much better in removing fog from the images. Unlike many DCP-based approaches that tended to remove fine details from the images along with fog, our study successfully avoided such unwanted effects, marking a notable achievement in preserving image details while mitigating fog.
- For LP identification, we use YOLOv8, a cutting-edge object detection framework. YOLOv8 was officially released on 10th January 2023 and when we started our investigation, it was a completely new algorithm with limited research.
- The study’s experimental findings show that the suggested model outperforms prior algorithms specifically developed for car plate identification in foggy environments. Our study demonstrates the higher performance achieved by the custom Bangla foggy dataset, the DCP-based dehazing technique, the execution of YOLOv8 and the utilization of EasyOCR via a comprehensive comparison.

1.6 Structure of Our Thesis Book

The thesis has been constructed in the following manner:

- In order to fulfill the objective of this study, in chapter 2, we discuss some of the noteworthy works from previous years, as well as some recent works that have been encountered about object detection and dehazing.
- Chapter 3 provides a comprehensive overview of the development of our work paradigm.
- Chapter 4 contains all the details of the datasets that have been used for conducting our study.
- Our detailed research methodology is elaborated in Chapter 5. Here, we have demonstrated how the models are constructed, the adjustments we made to get the desired parameters, how the model provides the desired output, and our entire workflow.
- In the first part of chapter 6, a detailed specification of the computer to run our algorithms is given. After that, qualitative and quantitative measurements demonstrate our model's superiority by comparing it with other models.
- Chapter 7 addresses some drawbacks of our work and suggestions for future study.
- The findings of our study, the significance, and overall analysis are discussed in Chapter 8.
- Finally, Chapter 9 concludes our thesis by summarizing it.

Chapter 2

Literature Review

This section discusses some of the notable works from previous years as well as some of the current work that has been encountered recently, which is helpful for our study to complete our goal.

2.1 Basic Object Detection Models

LP reading in emergencies is not new. Many studies have been done to figure out how to recognize a predicted LP area in a picture or video. Their ALPD techniques are built on top of many other methods, which are methodologies, systems, and processes for image processing. Many things, like the LP's shape, texture, and color are used alone or in combination to spot an LP area in a picture [7].

Abdullah et al. [25] conduct a study on Bangla LPR based on YOLO in Dhaka city. They used the YOLOv3 model to find number plates. The number plate in an image can be found and restricted using the object detection algorithm YOLOv3. The algorithm predicts bounding boxes and object classes for each grid cell by dividing the input image into a grid [55]. The researcher trained YOLOv3 to recognize Bangla LPs by making the algorithm fit the job and using pre-learned weights for the darknet-53 convolutional layers. Again, the authors employed YOLOv3 to identify the single Bangla number plates that have a character written in Bangla at the end of the first row and six numbers in the bottom row. Because it was very accurate and worked in real time, YOLOv3 was an excellent option for number plate detection in both situations. In order to make the process quicker without compromising performance, they dropped the vehicle detection step, which makes their approach significantly different from the approaches mentioned above. They had not encountered the issue of false positives in LP detection because they had not used SVM. This made their method effective because they avoided intermediaries by using YOLOv3 directly for LP detection. Unfortunately, the problem with YOLOv3 is that it did not work well with single objects, which can be hard to find. This is because YOLOv3 only has one scale for its feature map, which makes it hard to find things of different sizes. Also, YOLOv3 had trouble finding items that were partly hidden or had complicated backgrounds, which can cause false detection or recognition [47].

LP detection, character position, character extraction process, and character identification are only some of the five processes accomplished by Islam et al. [36]. They

introduced an approach to identify the ROI in the input picture. After using a changing boundary to get rid of the horizontal and vertical histogram values, ROI was taken from a picture of a LP by using many physical factors; such as size, containing box and aspect ratio, to find the histogram values. Linked component analysis and bounding box technology were used to break up the characters. The method may be used to locate both textual and non-textual objects. As a result, non-textual components were filtered out using an area and aspect based strategy. Character identification was done by feeding the extracted histogram of oriented slope (HOG) features into the support vector machine algorithm. SVMs can be hard to run on a computer, especially when they have to deal with big datasets or feature spaces with a lot of complicated features. This can slow down the time it takes to train and test, and it might not be good for real-time image processing [36].

On the other hand, Azam and Islam [13] provide a novel ALPD approach for identifying LP regions from images taken in hazardous circumstances. Using the frequency domain mask, they used a unique approach to remove rain patterns from images. Also, they used a novel strategy for increasing contrast using statistical binarization. They suggested a method for copying texts from images with low contrast and clarity or images taken at night or in foggy weather. They applied the Radon transform-based tilt correction approach to rectify tilted LP for the first time. A new criterion based on picture entropy was used to get rid of places that are not LP. This paper faced some difficulty in finding vertical labeled LPs; fortunately, this is a problem that the YOLO algorithm can fix.

Another approach has been introduced in this research. The authors of this research suggested a two-type detection pipeline which was integrated in order to use the Vision API to achieve a speed of reasoning in real-time while maintaining consistent accuracy in the detection and recognition of patterns. The proposed methodology comprised a two-stage detection module that employed a haar-cascade classifier as a prompt mechanism to rapidly eliminate frames that lack LPs and a MobileNet SSDv2 detection model for precise LP detection. The system considers the temporal separation among distinct vehicles to store their outcomes separately. Therefore, real-time reasoning speed is achieved while upholding an excellent ability to identify and recognize things. This approach used minimal hardware to achieve real-time inference speed. The authors considered that Bangladesh lacks high-end GPUs, unlike earlier techniques. The architecture can be used in real-time and low-resource settings because all inferences are done on a CPU with only one thread. In this paper, four experiments determined the optimum ALPR pipeline. The first pipeline used the baseline detection model YOLOv3 tiny. For the second pipeline, the YOLOv3 minuscule model and a cascade technique as a wake-up mechanism to discard the formulation faster without LP were used. This approach achieved real-time performance but had poor detection rates and OCR accuracy. Instead of YOLOv3 tiny, the third pipeline employed a trained MobileNet SSDv2 model for backbone detection. The cascade component was eliminated because it failed to recognize distant LPs in videos. The standalone MobileNet SSDv2 improved identification (98.0%) but not the F1 score compared to the baseline and second pipeline. Like the second pipeline, MobileNet SSDv2 was used as the main detecting model in the end process, and the cascade component was added. This outperformed the standalone

MobileNet SSDv2 in FPS (27.2%) and F1 (60.8%). The cascade classifier's consistency improved OCR accuracy, even if the detection rate was lower (82.7%) than the third pipeline. The MobileNet SSDv2 and cascade classifier pipeline delivered real-time performance, a higher F1 score, and accurate OCR results, making it the best pipeline for ALPR scenarios [48]. There is a great deal of potential for improvement in the OCR system's preprocessing calculation, especially in those areas where the weather may have an impact on the data.

Neural network-based models for deep learning, especially convolutional neural networks (CNNs) have emerged as powerful tools for solving complex computer vision problems. Redmon et al. [12] introduced the YOLO (You Only Look Once) algorithm, which is a widely used CNN-based method for detecting and recognizing targets. The YOLO algorithm has been improved through various means, such as amalgamating it with alternative techniques and refining its backbone architecture. In recent years, YOLOv5 has gained attention for its precision, speed, and efficient model size, making it suitable for embedded devices. The paper's key contributions are the light deep learning techniques that have been suggested and the single forward computation that allows complete LP identification and recognition. An innovative attention mechanism is added to the YOLOv5 algorithm to improve the effectiveness and precision of LP localization. The classifier's feature parameters are changed to increase model accuracy and shorten training time. For LP recognition without pre-segmentation, the recognition network uses GRU (Gated Recurrent Units) + CTC (Connectionist Temporal Classification). The CCPD dataset is used to evaluate the suggested technique, which shows better recognition precision when compared to the earlier algorithms. The authors developed a recognition network that achieves character segmentation-free recognition for LP recognition using Gated Recurrent Units (GRU) and Connectionist Temporal Classification (CTC). By using this method, the network's convergence speed and recognition accuracy are increased while training time is dramatically reduced. The experimental results show that the suggested model works really well. Even in complex situations, the model produces adequate recognition results due to its strong stability and durability [63].

The LP recognition process included the use of Gaussian noise, Gaussian blur, salt-and-pepper noise, coarse dropout, and normalization. LP recognition software YOLOv3 was used to identify the plate in the selected picture. By YOLOv3, the box was predicted, and the cost function was measured. YOLOv3 predicted an object score using logistic regression, and the cost function was computed in a new way. While training the data, they found the targeted detection model after 27 epochs. The previously stored model was employed for LP detection, and the projected bounding box location was used for cropping LPs from photos. Then, segmentation was used to take a closer look at each individual character on the plate. Segmentation was completed by Otsu's thresholding algorithm. The Convolutional Neural Network (CNN) model, which is a deep learning classification model, was used to recognize the characters. In this research, 81% of the predictions were correct.

In another study, M. M. S. Rahman [33] used a convolutional neural network (CNN) to identify Bengali LPs. Firstly, in the main image, the LP was cropped. After that, digits and characters were sliced into 32 pixels by 32 pixels for each. Then, individual characters were cropped from the LP in different scales, translations, rotations, shearings, and so on. By doing so, they had better accuracy for Bengali LP de-

tection. They used CNN to process the images. CNN consists of three layers: the convolutional layers, the pulling layers, and the fully connected (FC) layers. Convolutional layers and pulling layers are being used to extract features.

After using a support vector machine (SVM) to detect the characters on the LPs, Nahlah M [28] employs the honeybee method to finish segmenting the characters. The experimental findings demonstrate a positive benefit in recognizing LPs but a low recognition efficiency. The latest common recognition algorithms skip the character segmentation step and instead combine LP detection and recognition.

The LP localization module developed by Hengliang et al. [63] uses the modernized YOLOv5 model to detect and extract LPs from images. The recognition network then uses GRU to assign labels to sequences and decode them. Each data point's loss function value is determined to deliver the recognized LP results after the GRU output matrix and GT text are sent to the CTC loss function. Based on experimental findings, it is clear that the improved YOLOv5-LSE + GRU + CTC identification of LPs model is superior to the baseline version. The enhanced model achieves an average accuracy of 98.98% and a frame rate (FPS) that is commensurate with the needs of engineering applications.

2.2 Advanced Dehazing Methods

To dehaze the indistinct images, a significant study has been conducted. One such algorithm is DCP, an approach to picture dehazing. To further understand what makes DCP-based dehazing algorithms so successful, a literature review was undertaken. The research was systematically organized and analyzed throughout the four stages. According to the findings, determining the environmental light and transmission map relies heavily on the dimensions of the local patch utilized to construct the dark channel. The estimate of ambient light is precise if a large local area is used, and the use of entropy may further increase the estimation's accuracy. Entropy indicates the chaos of a tiny area of the picture in this case. A flexible adjustment approach is required for an accurate transmission map estimate, and employing a hazy image as a reference enhances the transmission map. For determining the transmission map, the soft matting approach has been proven to be the most accurate. However, standard quality measurements based only on the dehaze image are unreliable for assessing the algorithm's effectiveness. The step-by-step instructions for the DCP-based dehazing algorithm are involving generating the dark channel, estimating ambient light, determining the transmission map, enhancing the transmission map, modifying brightness and color, and saving the dehazed picture, might be used as a roadmap for researchers developing successful picture dehazing algorithms. Overall, the study gives useful information on the critical factors and strategies to consider while building DCP-based picture dehazing algorithms [17].

Numerous studies in recent years have shown promising results in identifying Bengali LPs. The recognition of Bengali LPs has been studied by M. A. A. Nasim et al. [44]. LP identification used additive Gaussian noise, blurring, salt-and-pepper noise, coarse dropout, and normalizing. LP recognition software YOLO-v3 was used to identify the plate in the chosen picture. The cost function was calculated, and the box was projected, using YOLO-v3. YOLO-v3 computed a new cost function and

made predictions about item scores using logistic regression. The previously stored model was employed for LP detection, and the projected bounding box location was used for cropping LPs from photos. Then, segmentation was used to take a closer look at each individual character on the plate. Otsu's thresholding technique was used to finish the segmentation. A deep learning classification model called CNN was employed to decipher the characters. The accuracy rate for these forecasts in this study was 81%. However, it improved its performance after being trained with a large amount of data.

The recognition of LPs as part of automatic vehicle identification is a technology that may be relied on to accurately identify vehicles. A method that is both effective and efficient for recognizing the number plates on vehicles has been developed and implemented in this research. The goal of this research was to retain accuracy while overcoming scaling challenges as well as character position identification challenges. In this paper, they introduced five distinct phases to LP reading. To begin, a photograph of the LP from the vehicle must be captured and processed using an image recognition system. The second stage is image preprocessing, during which the image is adjusted for contrast, brightness, and other qualities. Finding and identifying the LP region in the image is the third phase, often known as LP localization. Separating the symbols on the LP into their constituent characters constitutes the fourth stage. In the final stage, characters are converted from their segmented form into text by optical character recognition (OCR) technology. However, the Font, size, color, and overall design of LPs can vary substantially. This variability presents a challenge for EasyOCR algorithms, which must accommodate numerous plate formats and regional standards. This can reduce the precision and dependability of character recognition and extraction from LPs. Moreover, EasyOCR algorithms significantly rely on the input image's quality. Low resolution, motion blur, noise, and reflections can reduce the legibility of the LP image, making it difficult for OCR algorithms to recognize characters accurately. Poor image quality can result in increased error rates and decreased efficacy overall [49].

Using several indices, the author compared various dehazing models. In his studies, Neural Network-based models performed much better in terms of PSNR and SSIM. However, statistical methods dehaze photos faster and make them seem sharper. Prior-based dehazing models, according to the author, seem better in human eyes because they keep sharpness, contrast, and lighting. The author stated that to get higher PSNR and SSIM, neural network-based models lose clarity. His investigations also showed that classical models help object detection algorithms to detect objects with a higher degree of accuracy on average. But despite being a Neural network model, MSCNN did quite well in tests, with higher PSNR and SSIM scores as well as very high assurance, although it is substantially slower than classical prior-based models [26].

According to He et al. [39], classical prior base models are more reliable and quicker than neural network-based dehazing algorithms. They also sought to enhance He's latest model by including quicker and more natural-looking pictures with higher SSIM and MSE scores. In order to avoid further enhancing the transmission map and hence reduce processing time, they recommended creating dark channels by adaptively fusing two typical channels measured in a double-scale window [39].

After using SVM to detect the characters on the LPs, Nahlah M. [18] employs the

honeybee method to finish segmenting the characters. The experimental findings demonstrate a positive benefit in recognizing LPs, but a low recognition efficiency. The latest common recognition algorithms skip the character segmentation step and instead combine LP detection and recognition.

Another paper titled “Night Video Enhancement Using Improved Dark Channel Prior” by Xuesong et al. [8] gives a novel approach to use DCP and enhance the light of the video for enhancing quality in low lighting conditions and poor visibility. To improve the details, researchers used local smoothing on the DCP algorithm and image Gaussian pyramid operators. The comparison between Dong et al.’s [6] method and their approach is major: Dong’s method does 23 frames per second, Xuesong’s approach does 33. the smoothing technique using a median filter on a coarse dark channel map (DCM) to obtain a pixel-wise map and noise reduction.

When we looked for a better approach for dehaze images, we found a paper by Chuan et al. [58]. DCP, BCCR, and Ehsan findings show color distortion. The sky section of NonLocal is overexposed. The sky is distorted in MSCNN, DehazeNet, AODNet, and He, while the road is excessively dark in DehazeNet, AODNet, He, Ehsan, and D4. DehazeFormer-T and gUnet-T restorations are better, but they leave more residual haze. Their approach lowers distortion and removes haze. We measured haze using a fog aware density evaluator (FADE) [11], as there is no GT picture to quantify the PNSR/SSIM index. The FADE index score indicates haze density.

On the other hand, we found another paper by Yash et al. [62] where they suggested using OCR and a new method based on deep learning to automatically find and read number plates. Using deep learning, the model is taught to spot the car. The part of the picture that shows the LP is cut out well, and a convolutional Neural Network (CNN) uses OCR to figure out what the numbers and letters are. The Jetson TX2 NVIDIA target was used to train the model, and a public dataset from the Kaggle database was used to test how well it worked.

Besides, Wang et al. [29] introduce a single picture dehazing technique based on a physical model and image brightness components to resolve foggy weather photographs. Its greatest contribution is its capacity to dehaze a single picture by estimating the ambient light value and transmission map while accounting for the image’s dynamic range. Misty weather causes visual deterioration, which the author discusses in detail. Haze causes diminished vision, contrast, and color distortion due to airborne particle scattering, refraction, and reflection.

Moreover, Pazhani and Periyanyagi [52] focus on the Retinex algorithm, a popular haze mitigation approach. It underpins the URSHR algorithm. Their study reveals that urban remote sensing algorithms are parameter-sensitive and limited. A long-lasting urban remote sensing system is required. They describe the unique and well-reasoned use of phase information from remote sensing pictures with the Retinex theory to reduce haze. Their method reduces haze and retains the scene’s vivid features, according to the authors. A distinctive contribution to the region has great potential. The research shows URSHR’s broad image type support, effective haze reduction, and increased edge details. Complex photos are analyzed effectively by the system, demonstrating its limits. This study presents a well-organized and informative investigation of urban remote sensing haze removal. URSHR, which

combines novel methodologies with a profound understanding of existing procedures, may overcome this challenge. Positive results from extensive testing improve the paper's credibility.

Chapter 3

Background

Our work is divided into three sections: firstly image defogging using DCP, followed by object detection using YOLOv8 and object recognition using OCR. We require some prior information to ensure the procedure and alter the environment in order to comprehend the methodology. This part gives a thorough understanding of the history of our work paradigm.

3.1 Dehazing with Dark Channel Prior Method

Under hazy atmospheric conditions, this detection process may encounter challenges attributable to particulate matter and fog. These adverse conditions often result in reduced intensity levels of the LP or a diminished dynamic range, leading to the creation of shadow-like artifacts that adversely affect detection accuracy. Hence, a critical prerequisite for LP recognition in such scenarios involves an initial data preprocessing step.

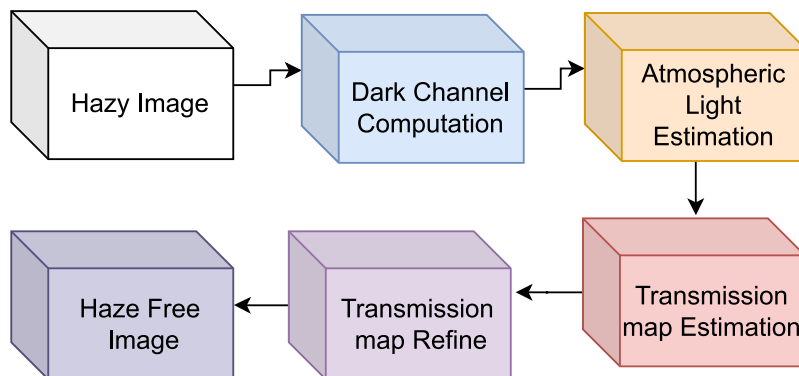


Figure 3.1: Dark Channel Prior Architecture

The data preprocessing operation is fundamentally instrumental in rendering the LP more discernible within the image. In this context, DCP has been employed as a mean to address dynamic range limitations and enhance image contrast. The DCP algorithm commences by segregating the input image into its constituent color channels, specifically the green (G), blue (B), and red (R) channels. Subsequently, it calculates the dark channel, which corresponds to the channel featuring the smallest pixel values. To further augment the distinctiveness of the dark channel, a small square kernel is applied through an erosion operation. This preprocessing step serves

to accentuate the LP within the image, rendering it more prominent and thereby facilitating subsequent recognition tasks.

3.1.1 Atmospheric Light Analysis

According to He et al. [3], estimating the atmospheric light (A) can be done with a subset of the darkest pixels in the dark channel. It computes the average value of the darkest pixels in the dark channel and the brightest pixels to measure the light, as the atmospheric light has the highest intensities from those pixels. This average serves as the atmosphere's illumination. A_i is specified in equation 3.1.

$$A_{i=R,G,B}^{n+1} = \frac{1}{2} (A_i^n + R(x_n + 1)) \quad x \in \tilde{R} \quad (3.1)$$

where \tilde{R} is the order of the top 0.2% of the brightness value in the dark channel of $R(x)$. The value of A_i is found by comparing and updating the average of the pixel points that are next to each other in \tilde{R} . Because of this, each repetition must be compared to t . Through repetition, the places where the dark channel brightness is not very noticeable are taken into account.

3.1.2 Transmission Map Forecast

The transmission map (t) is calculated using the atmospheric light channel and the dark channel. Here, $\tilde{t}(\mathbf{x})$ in equation 3.2 is determined by the expression.

$$\tilde{t}(\mathbf{x}) = 1 - \min c \left(\min \mathbf{y} \in \Omega(\mathbf{x}) \left(\frac{I^c(\mathbf{y})}{A^c} \right) \right) \quad (3.2)$$

where I^c is the normalized input image divided by the atmospheric light A^c and omega is a constant. In fact, $\min c \left(\min \mathbf{y} \in \Omega(\mathbf{x}) \left(\frac{I^c(\mathbf{y})}{A^c} \right) \right)$ is the dark channel of the normalized haze image $\frac{I^c(\mathbf{y})}{A^c}$. It directly provides an estimation of the transmission.

3.1.3 Transmission Refinement via Guided Filtration

According to He et al.[3], using directed filtering to refine the transmission map enhances edges and maintains essential details. As inputs, the guided filter utilizes the gray-scale image and the estimated transmission map to generate a refined transmission map (t). $t(\mathbf{x})$ stands for the refined transmission map. We minimize the following cost function by writing $t(\mathbf{x})$ and $\tilde{t}(\mathbf{x})$ as \mathbf{t} and $\tilde{\mathbf{t}}$ in the form of a vector.

$$E(\mathbf{t}) = \mathbf{t}^T \mathbf{L} \mathbf{t} + \lambda (\mathbf{t} - \tilde{\mathbf{t}})^T (\mathbf{t} - \tilde{\mathbf{t}}). \quad (3.3)$$

The equation 3.3 is the cost function, where \mathbf{L} is the Matting Laplacian matrix that had been proposed by Levin. Here, λ is a regularization parameter.

3.1.4 Image Restoration

The process involves the reconstruction of the dehazed image through the utilization of the improved transmission map (t), the atmospheric light (A), and a conservative threshold value. Adjustments and shifts in pixel values for each color channel are

carried out to regain the dehazed image (J), a technique described by He et al. [3]. Upon examination of the intensity frequency distribution before and after the application of the DCP, it becomes evident that the dynamic range has undergone alteration. Specifically, post-application of the DCP, the intensity values exhibit a more dispersed distribution compared to the initial state, wherein the image displayed greater prominence.

3.2 Detecting Object Using YOLOv8

The YOLO algorithm was used to detect objects because it is highly efficient and can run at real-time speeds even on low-end hardware. Specifically, YOLOv8 was used here. By incorporating new techniques YOLOv8 has improved object detection accuracy. From any other object detection models YOLOv8 has faster inference speed. As this model supports multiple backbones, the users may choose the most suitable one for them. In figure 3.2 we can see the detailed architecture of YOLOv8.

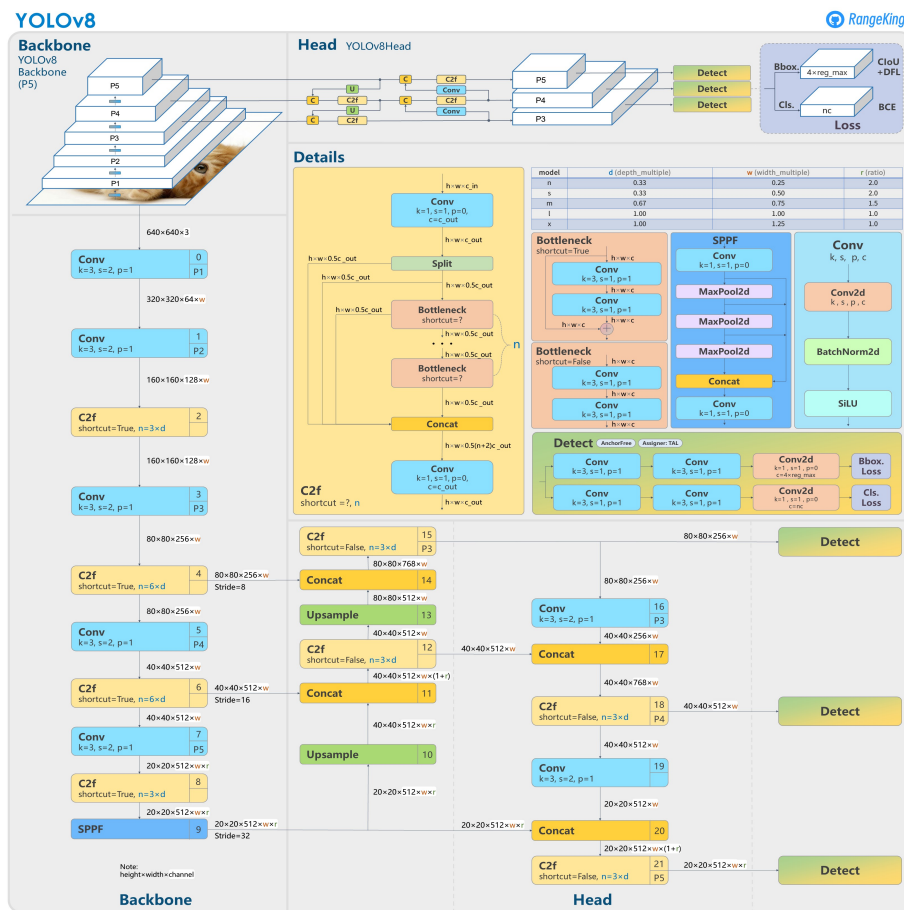


Figure 3.2: Model Structure of YOLOv8 [64]

The basic architecture of YOLOv8 was built upon the previous versions of the YOLO algorithm in figure 3.3. This model basically has a convolutional neural network which has two main parts and they are:

- The head
- The backbone

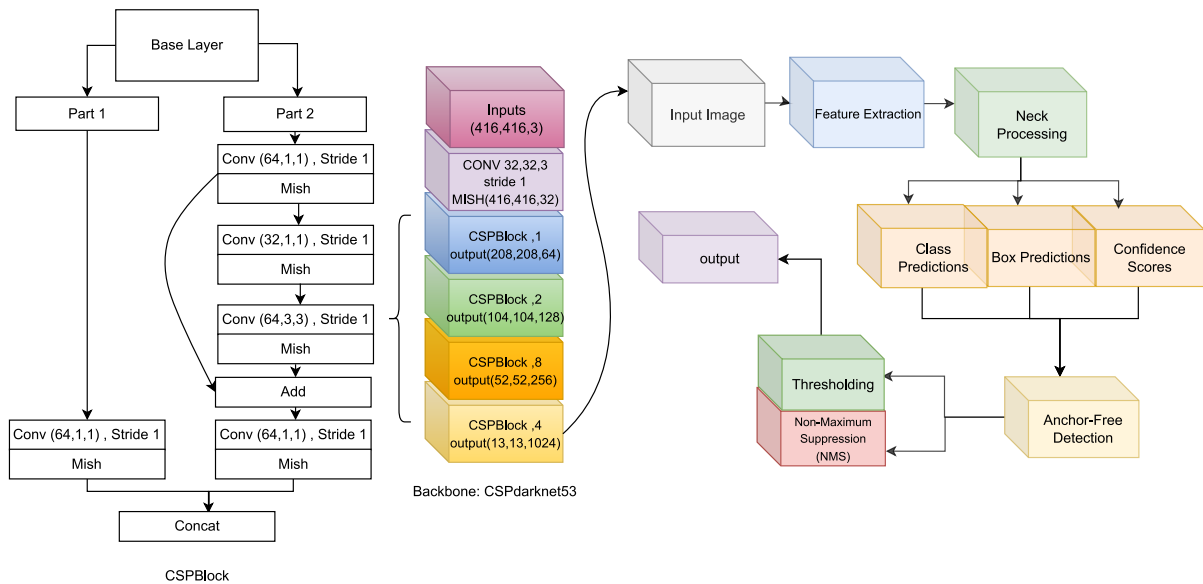


Figure 3.3: The Structure of YOLOv8 With Backbone

A number of convolutions layers and a string of fully connected layers make the head of the YOLOv8 algorithm. These layers predict bounding boxes, objectness scores and class probabilities for the recognized objects in an image. By using the self-attention mechanism in the head, this model is able to focus on different parts of the image and adjust the importance of different features according to their relevance to the task. The backbone of YOLOv8 is formed by a modified version of CSPDarknet53 architecture. This architecture has 53 convolutional layers. To improve the flow of information between those layers cross-stage partial connections are being used. Multi-scaled object detection can also be performed by YOLOv8. In order to find items of various shapes and sizes within a picture, this model uses a feature pyramid network [35], [59].

3.3 Recognition Using OCR

The abbreviation “OCR” describes the process of recognizing text visually. It is a system that can read and extract texts from scanned documents or photos. Characters in an image are deciphered by OCR software by assessing its geometrical properties and then translating those properties into text that can be read by a computer. OCR relies heavily on a synthesis of computer vision and machine learning methods. In figure 3.4 we can see the overall workflow of OCR.

Here, we used EasyOCR which is a Python package that uses PyTorch as its back-end handler [41]. It provides a simple and accurate solution for identifying text from images. It is based on the well-known OCR engines Tesseract and Kraken, and it offers a high-level API for OCR operations. It is the simplest method for extracting text from photos, and it is much more reliable when a powerful deep learning library (PyTorch) is backing it in the background. EasyOCR allows users to extract text from images with just a few lines of code. [57]

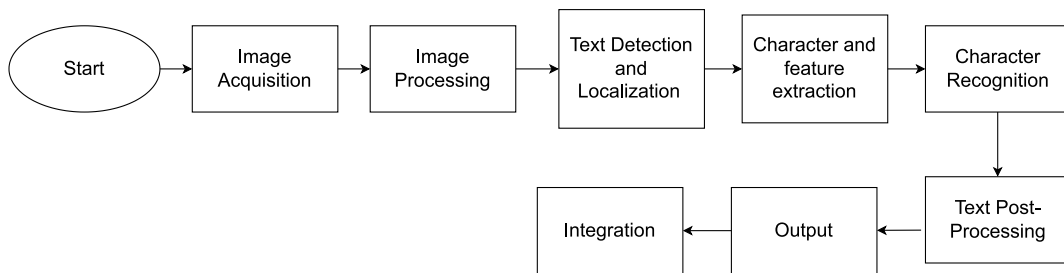


Figure 3.4: OCR Workflow

The accuracy of EasyOCR is one of its key features. It recognizes text with great accuracy using powerful OCR engines and has been tested on a wide range of real-world photos. Furthermore, for detection purposes, EasyOCR supports 42+ languages including English, Bengali, Spanish, German, and French making it a versatile tool for a variety of applications. EasyOCR was developed by the Jaided AI company [57] Here are some characteristics of EasyOCR:

1. EasyOCR software typically has a sample that allows users to effortlessly import photos, begin the OCR process, and view or export the recognized text.
2. These tools frequently offer automated image preprocessing and text recognition, reducing the need for users to set up complex settings or make manual adjustments.
3. EasyOCR generally handles a wide range of image formats, such as scanned documents, PDFs, and image files (e.g., JPEG, PNG), making them adaptable to a wide range of input.
4. Even though simplicity is a key feature, EasyOCR still manages to recognize text correctly. A lot of these options use advanced OCR algorithms to improve accuracy.

5. It supports many languages, allowing users to recognize text in a variety of languages and character sets.
6. Users can often export recognized text to a variety of formats, including plain text, Word documents, and searchable PDFs.
7. Easy OCR provides integration with other software or platforms, making it simple to incorporate OCR capabilities into existing workflows.

3.3.1 Comparison Between EasyOCR and Other Alternative Techniques for Text Recognition from Images

EasyOCR is distinguished as a robust text recognition tool by a number of important benefits. It automates the extraction of text from images, drastically reducing the need for manual labor and saving valuable time, making it ideally adapted for the efficient management of large volumes of text-based images. In contrast, manual data entry is time-consuming, error-prone, and labor-intensive, rendering it unsuitable for large-scale duties. EasyOCR has specialization in recognizing textual content within images, while object detection and extraction techniques excel at identifying and isolating non-textual objects or regions. Moreover, EasyOCR extracts text directly from images, enabling machine-readable text for keyword searches, whereas keyword search techniques are primarily concerned with locating images using specific keywords associated with them. In addition, EasyOCR provides automated and systematic text extraction, thereby it eliminates the need for manual image evaluation and tagging. In conclusion, EasyOCR is a top choice for accurate and efficient text recognition from images and scanned documents, particularly when structured or unstructured text extraction is the primary objective and it stands out as an approachable alternative for those without extensive technical knowledge or programming experience.

Chapter 4

Dataset

Here we would like to provide two secondary datasets that have been customized using monocular depth to create a foggy image as there is no dataset having a foggy Bengali car plate. These datasets will display the characteristics and variety, which might assist in discovering little features that are not visible in small or non-variable datasets.

4.1 Dataset Analysis

We have used two secondary datasets, one with 2,754 samples [46] and the other with 502 samples [38]; the training set uses 70% of dataset-1 (1928 photos) and dataset-2 (351 photos), the test set uses 15% of dataset-1 (413 images) and dataset-2 (75 images), and the validation set uses 15% of dataset-1 (413 images) and dataset-2 (75 photos). The pictures span the whole range of human perception in terms of sharpness and clarity, distance and proximity, number plate attachment and detachment, in daylight and nighttime, and other aspects. The dataset-1 contains 46.6% automobiles, 2.6% only LPs, 39.4% motorcycles, 3.3% compressed natural gas (CNG), and 8.1% buses and trucks. Alternatively, in dataset-2, 70.11% are automobiles, 9.56% are pickups, 9.56% are bikes, and 10.77% are buses.

Vehicle Type	Dataset-1 (%)	Dataset-2 (%)
Automobiles	46.6	70.11
License Plates Only	2.6	-
Motorcycles	39.4	9.56
Compressed Natural Gas (CNG)	3.3	-
Buses and Trucks	8.1	10.77
Pickups	-	9.56

Table 4.1: Comparison of Vehicle Types in Dataset-1 and Dataset-2

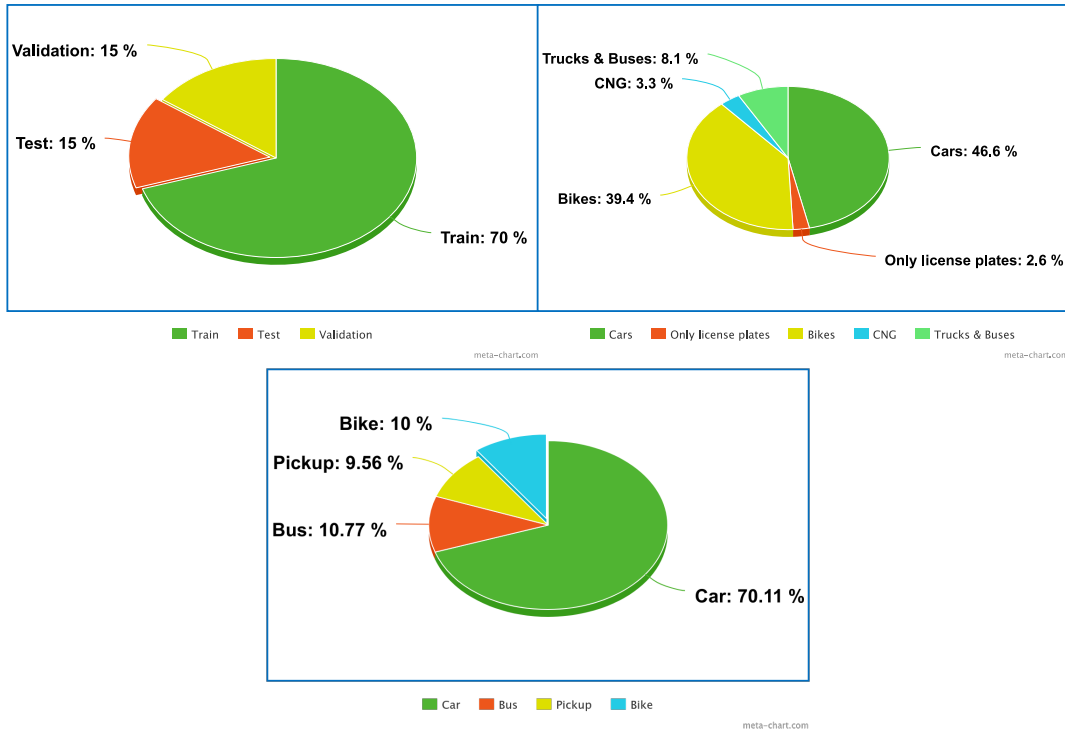


Figure 4.1: Dataset Split and Class Variety Visualization

4.2 Customised Dataset

We opt in to work with customized datasets. For that, firstly we took secondary data [38], [46] to manipulate and modify. In our dataset-1 [46] there are two parts of this. In the first part, there are all images of cars with LP. In the second part of this, there is only a picture of an LP. As we are removing environmental hazards and detecting the part of the number plate is our main concern thus we decided to use the first part and for dataset-2 [38] we are using all the images. The first part has 1928 images of the vehicle for training, 413 for validation, and 413 for testing, and in the second part, 502 images have been taken in concern. We chose to modify these pictures, the dataset is already in YOLO format thus it will be beneficial for us to use. Creating a foggy effect in images typically involves simulating the reduction in visibility caused by atmospheric conditions like fog.

We have used monocular depth estimation to create fog in the images. More than 3000 foggy images are synthesized using monocular depth estimation. By this, the scale of the dataset is expanded to solve the problem of insufficient Bangla car plate dataset under foggy weather. In this paper Monodepth2 Network proposed by Godard et al.[32] is used for depth estimation of the clear images in the Bangla car plate dataset, which ultimately generates the foggy images. According to a paper by Xiangyu Nie [51], Monodepth2 can achieve exceptional depth estimation by using the lowest reprojection loss, full-resolution multi-scale sampling approach, and auto-masking loss. It can be trained with either monocular or binocular pictures as input data. Since our goal is to create foggy car plate images with varying densities based on distance. So, getting the actual and precise depth information is not required. Instead, the trend of depth variation or the relative fluctuation of depth fulfills the criteria. Therefore, Monodepth2 has been used for this research.

A prediction of the depth to get the fog effect that was wanted, the mono-depth model had to be used with a number of important steps. To start, we need to choose a good depth measurement model that has already been trained to figure out how far away things are in photos. Once the model is in place, the next step is to load the picture we want to use for the fog effect.

After importing the picture, we can use the model which has already learned to make a depth map of the picture. This depth map is an important part of duplicating the fog because it shows exactly how far away different things in the shot are from the camera.

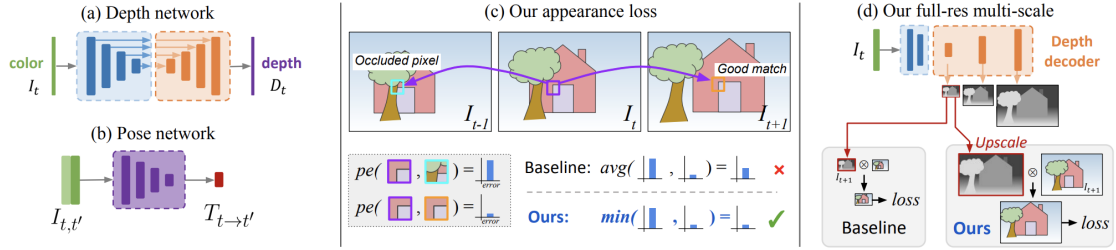


Figure 4.2: Approach Combines a Depth Network, Pose Network, Per-Pixel Minimum Reprojection Loss, and Full-Resolution Multi-Scale Processing for Enhanced Depth Estimation [32].

In a research paper referred to “Digging into Self-Supervised Monocular Depth Estimation”, written by Godard et al. [32] described a way for a trained person to estimate the depth of an object using only one eye. In self-supervised training, the model learns to guess the picture from the point of view of another image. It does this by using the difference as an intermediate variable, from which we get the depth map (Dt). Subsequently, an error is quantified according to Equation 4.1 to ascertain the final image and the associated generation process [32].

$$L_p = \min pe(I_t, I_{t' \rightarrow t}) \cdot t' \quad (4.1)$$

where L_p is the mistake in photometric projection and pre-photometric pe reconstruction error, which is calculated using SSIM and L1 distance.

SSIM and L1 distance are both parts of the photometric restoration mistake, which we get from various research works [1], [19], [23]

$$pe(I_a, I_b) = \frac{\alpha}{2} (1 - \text{SSIM}(I_a, I_b)) + (1 - \alpha) ||I_a - I_b|| \quad (4.2)$$

In Equation 4.2, $\alpha = 0.85$ is the weight of the wall, as in we use edge-aware smoothness. A smoothness term (equation 4.3) has been added to ensure the depth maps are smooth.

$$L_s = |\partial_x d_t| e^{-|\partial_x I_t|} + |\partial_y d_t| e^{-|\partial_y I_t|} \quad (4.3)$$

According to Godard et al.[32] various techniques, including pre-pixel minimum re-projection, have been employed to enhance the precision of the projected depth maps while refraining from introducing additional models into the framework. To address scenarios involving moving objects, the minimum error and auto-masking stationary strategies were implemented. The binary mask μ in equation 4.4 is dynamically

computed, taking into account projection errors and a multiscale estimation, to prevent convergence to local minima during the optimization process. The collective training loss is obtained by aggregating the per-pixel smoothness term (L_s) with the filtered photometric loss [22], [24], [27].

$$\mu = \left[\min_{t'} pe(I_t, I_{t' \rightarrow t}) < \min_{t'} pe(I_t, I_{t'}) \right] \quad (4.4)$$

As shown in figure 4.2 we derived the depth maps from the original images. Subsequently, we inverted these images. Once the model is appropriately configured, we proceed to import the designated target image onto which we intend to simulate the foggy conditions. We also explore methodologies to improve the accuracy of the projected depth maps without the need for additional model complexity. Upon meticulous examination, it becomes evident that the artificially generated foggy images closely resemble their corresponding original counterparts, exhibiting minimal discernible distinctions.

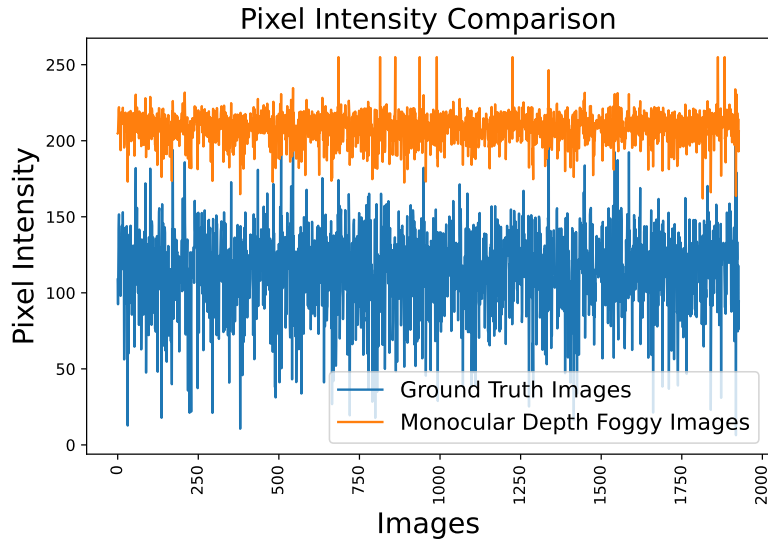


Figure 4.3: Mean Intensity of Ground Truth Images and Customized Foggy Images

In a conventional haze-free image, the color image comprises pixel values ranging from 0 to 255, serving as indicators of the clarity of each pixel’s coloration. An image of moderate clarity tends to exhibit greater prominence of both bright and dark values. Conversely, in cases of foggy imagery, there is an observable prevalence of brighter pixel values and a corresponding decrease in darker values due to the limited visibility caused by the fog. As depicted in Figure 4.3, it is noteworthy that the mean intensity value of the ground truth image is lower than that of the monocular depth-based foggy image. This discrepancy serves as evidence affirming the presence of authentic fog within the customized dataset.

Following the acquisition of the depth-affected image under foggy conditions, we proceed to amalgamate this depth image with the corresponding clear image. This fusion process is subject to the regulation of fog density levels, governed by a beta regulator. As depicted in Figure 4.4, the illustration showcases individual images exhibiting varying mean intensity values across distinct beta settings. Notably, at a beta value of 7, the image demonstrates a maximum intensity value that notably

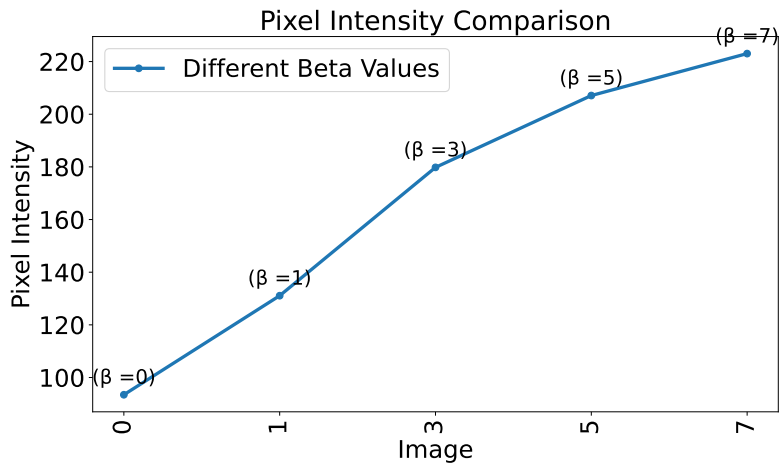


Figure 4.4: Variation in Density Mean Intensities of the Individual Image

exceeds 200, whereas the authentic image typically exhibits mean intensities within the range of 50 to 100.

Since we can not discover any secondary dataset and owing to restrictions in the environment and condition of the current circumstances, we would like to work with real-world datasets in the future after gathering primary data.

Chapter 5

Research Methodology

We have shown the models we want to encounter in the background; now we want to demonstrate how we built the models, what changes we make to acquire the parameters we want, and how the models produce the outputs we want and our overall workflow.

5.1 Workflow

First of all, the hazy picture that was provided will be dehazed using the DCP technique. This will dehaze the image and make the number plate's intensity stand out more prominently. Following this, a trained model of YOLOv8 will be used to locate the region of the picture that contains the real number plate. Then, using a technology known as OCR, we will find each and every character. Once it is complete, the LP will be retrieved. Following the completion of the fetch operation, we will evaluate how well our model is functioning and identify any remaining gaps in its coverage. Finally, the overall procedure is shown in 5.1.

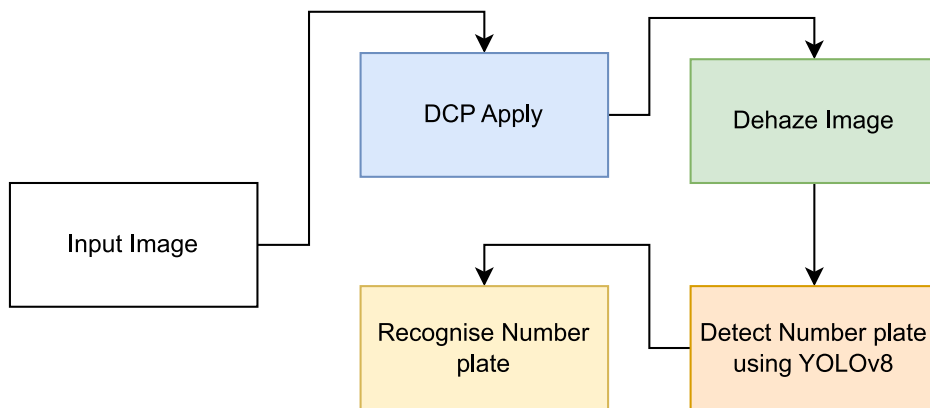


Figure 5.1: Work Flow

5.2 Data Pre-Processing Using DCP

In order to demonstrate the dehaze picture and improve the overall quality of the image, the image under consideration was initially generated at three distinct scales. One is the original, another is a half-size version, and the final one is a scaled-up version that is twice as large as the original.

At this point, each and every image will be processed by the algorithm for the typical dark channel. In the first place, we need to take measurements of the dark channel. The morphological kernel of 15×15 has been utilized for the purpose of calculating the dark channel, and the minimum value from each channel in a color image has been taken into consideration.

Following the estimation of the dark channel, we have done the computation of the ambient light by utilizing this channel. For the purpose of calculating the transmission map, just a small percentage of the light from the atmosphere has been taken into consideration.

The value of the ω has been set to 0.80 for the transmission map, and we have computed the real transmission map by selecting the dark channel as the channel of interest shown in figure 5.2. The ω contributes to the scalability factor, which is essentially less than 1. We maintain it at 0.80 in order to maintain its natural appearance and to avoid the emergence of artifacts.



Figure 5.2: Dark Channel Image and Transmission Map Forecast Image

Following the completion of the estimation of the transmission map, we will need to tweak it. We have utilized a guided filter in order to improve or smooth out the edge's appearance. The hyper parameters for the process are as follows: 1) epsilon,



Figure 5.3: Transmission Refinement via Guided Filtration and Image Restoration

which is a regularization that prevents division by zero and controls the degree of smoothing; 2) radius, which indicates what the size of the guided filter will be. With the use of the guided filter, we have been able to predict the transmission refinement, which was crucial in obtaining the dehaze image through the use of refactor and the final restored picture after pre-processing shown in figure 5.3.

Following the completion of all the pre-processing steps depicted in figure 5.4, we are now able to figure out the overall step.



Figure 5.4: Pre-processing Using DCP

5.3 Proposed Model

While the hybrid technique in figure 5.5 only applies the conventional dehazing operation to a single photo, our hybrid DCP applies it to a three-scale image. This is in contrast to the hybrid method, which only operates on a single picture. There are three distinct sizes available: the original, the half-scale, and the double-scale. The images are processed on a variety of scales, which enables our scale to capture medium and small details that could be lost on a single scale. Processing at a lower scale enables better handling of small features, but processing at a larger scale may lead to a larger hazy zone. Processing allows for better handling of small features. Following the dehazing of each scale, our objective was to combine the various resolutions that had been dehazed by using a technique known as multi-scale fusion to combine these images.

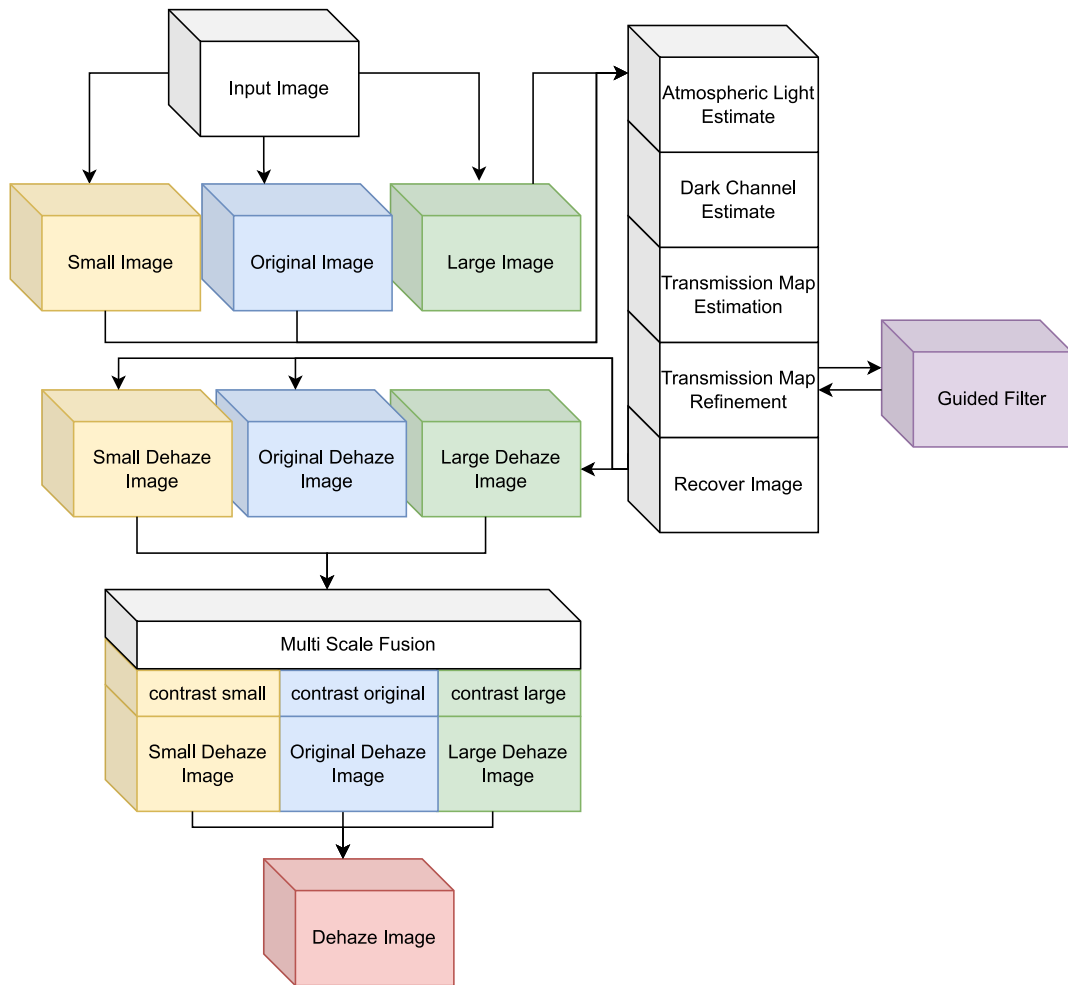


Figure 5.5: Proposed Model Architecture

The multi-scale camera captures images that have been processed at several sizes, with the scale being determined by the amount of contrast on the image. The contrast has been determined by using the laplacian operator [2], which is a measure of the sharpness or edge intensity of an image. The input images are weighted according to their contrast in order to construct the final fused picture. A higher contrast means that there is less haze and that the information is clearer. A weighted

average that is based on the approach will be utilized to achieve balance. This will be accomplished by using the smoother and less noisy quality that is typical of low-contrast images.

Proposed Algorithm

Input: List of image filenames

Output: Dehazed images

- 1: *Initialization:*
 - 2: Set $\omega = 0.80$
 - 3: Set $\varepsilon = 0.001$
 - 4: Set radius $r = 105$
 - 5: Set structuring element size $s = 15 \times 15$
 - 6: **for** each image filename in the list **do**
 - 7: Read the hazy image.
 - 8: Normalize the image: $I = \frac{\text{src}}{255}$
 - 9: Compute the dark channel: $J_{\text{dark}} = \min(\min(R, G), B)$
 - 10: Apply erosion with kernel of size s to J_{dark} .
 - 11: Estimate A using top 0.1% brightest pixels in J_{dark} .
 - 12: $t = 1 - \omega \times J_{\text{dark}}$
 - 13: Convert image to grayscale I_{gray} and normalize.
 - 14: Refine t using guided filter with radius r and ε .
 - 15: Recover the scene radiance: $J = \frac{I-A}{\max(t, t_0)} + A$
 - 16: Resize the image to different scales and apply dehazing.
 - 17: Fuse the dehazed images from different scales.
 - 18: Save the final dehazed image.
 - 19: **end for**
-

$$\nabla^2 I(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 I(x+i, y+j) \times \text{Laplacian Kernel}(i, j) \quad (5.1)$$

Here in (5.1) the mathematical operator ∇^2 is utilized in image processing for edge recognition and is called the Laplacian operator.

When dealing with images that are represented by pixels in their discrete form, a common approach is to use a convolution operation with a Laplacian kernel to approximate the Laplacian operator. In discrete Laplacian theory, the Laplacian kernel is a 3x3 matrix and $I(x, y)$ represents the pixel intensity at coordinates (x,y), and the corresponding equation is (5.1).

Using fusion and multi-scale picture dehazing, we are able to demonstrate the hybrid algorithm's ability to dehaze and enhance the visual contrast of the image in figure 5.6.

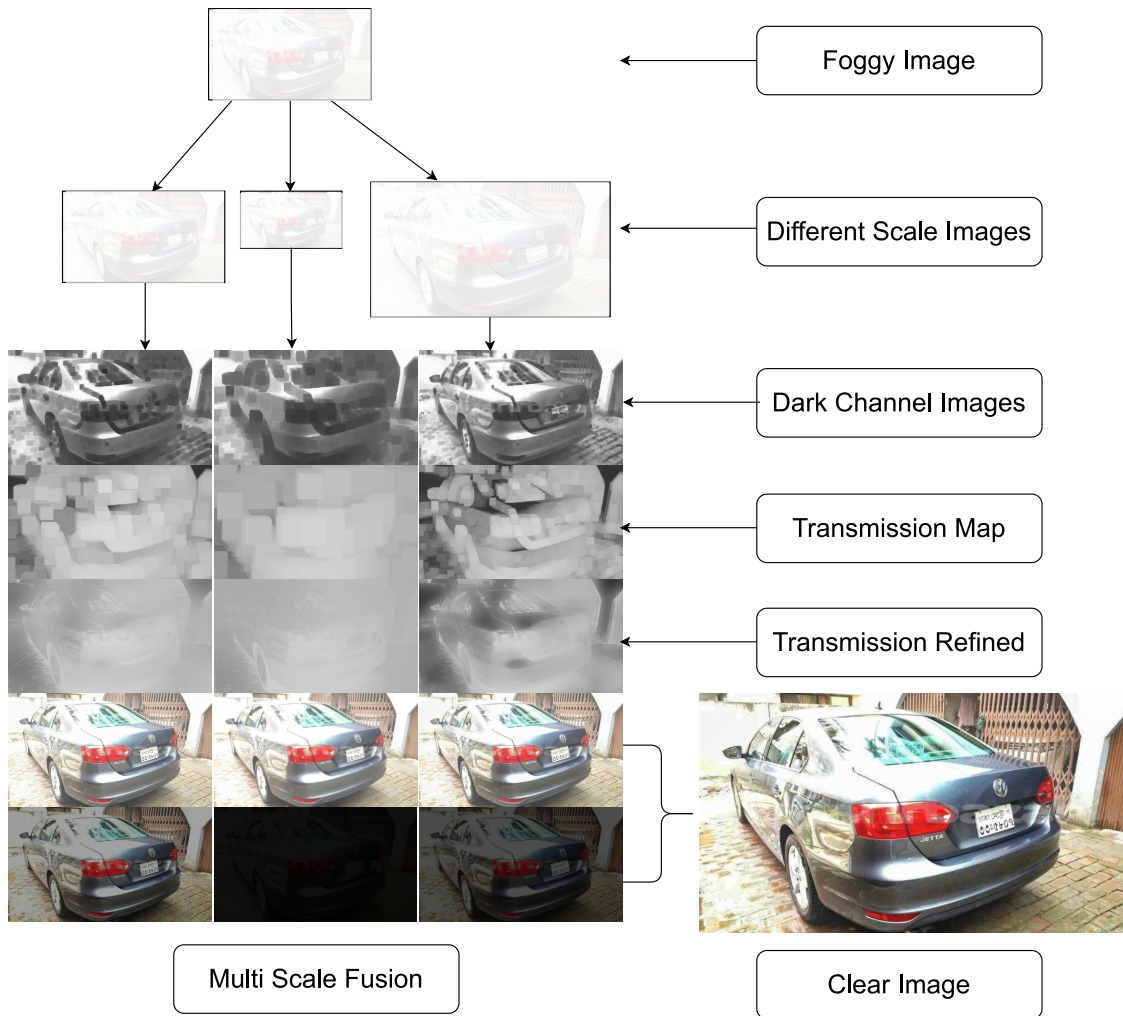


Figure 5.6: Proposed Model Working Procedure

5.4 Detection Using YOLOv8

In this segment we will discuss the workflow of YOLOv8 step by step. So, if we look figure 5.7 we can see the overall procedure.

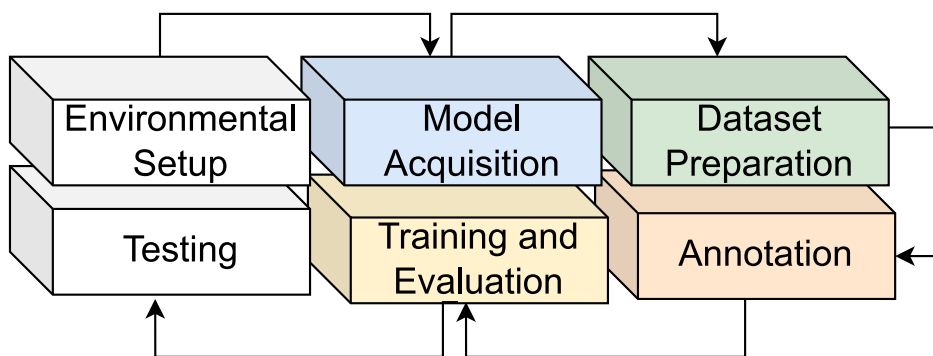


Figure 5.7: YOLOv8 Workflow

1. Data Collection:

In our work, we've used a customized dataset and almost 2754 pictures were used in our dataset. As we are dealing with hazy weather conditions, we took those pictures that were hazy. Then the DCP algorithm was used to dehaze the pictures by removing the fog and rain.

2. Data Annotation:

Then we annotated the data by labeling each number plate in every picture so that it can be recognized by the YOLO model. In order to annotate the data an online tool CVAT was being used.

3. Dataset Structure:

As YOLOv8 needs the data in a specific format which is the YOLO 1.1 format. So we have to convert all those annotated images from CVAT into YOLO 1.1 format so that we can input these data into the YOLOv8 model. For this we had to choose YOLO 1.1 as the export format and downloaded all the annotated images in the YOLO 1.1 format. All the annotated images were downloaded as text files. All the text files represented each bounding box with 5 parameters within a single row. Here the first parameters indicate the class, the second and third parameters represent the center of the bounding box, the fourth parameter is the width of the bounding box and the fifth parameter is the height of the bounding box. For every single image in the image directory we had an annotation file in a labels directory which has the exact same name with .txt extension [56].

4. Training:

Google Colab was used to train our model. Ultralytics library was used to import YOLO. Here, we used the "m" version of YOLOv8. The yaml file contains all the configurations for our training. In 'names' all the different classes will be set here. As we are detecting only number plates, so there will be only one class. 'path' contains the absolute path to the directory that contains the images and the labels. 'train' also gives a directory, and it is related to the path. The data that the algorithm will use for training purposes is located in this directory. 'value' is basically the validation dataset. It also gives a directory, and it is related to the path. The value of the epoch was 100.

All training measures (box loss, classification loss, and distance focus loss) go down over time, showing that the model is learning well. With a mAP50 of 0.6 and a mAP50-95 of 0.4, the validation measures show that the model also works well on data it has not seen before.

- (a) Train/box_loss: This statistic determines how accurate the projected bounding boxes are compared to the actual bounding boxes on the ground.
- (b) Train/cls_loss: This metric determines how far off the predicted class labels really are in comparison to the ground truth class labels.
- (c) Train/df_l_loss: This metric measures the distance focal loss, a weighted loss function that helps improve the performance of object detectors on small and difficult objects.

- (d) Metrics/precision(B): This metric measures the precision of the model, which is the fraction of detected objects that are actually present in the image.
- (e) Metrics/recall(B): This metric measures the recall of the model, which is the fraction of ground truth objects that are detected by the model.

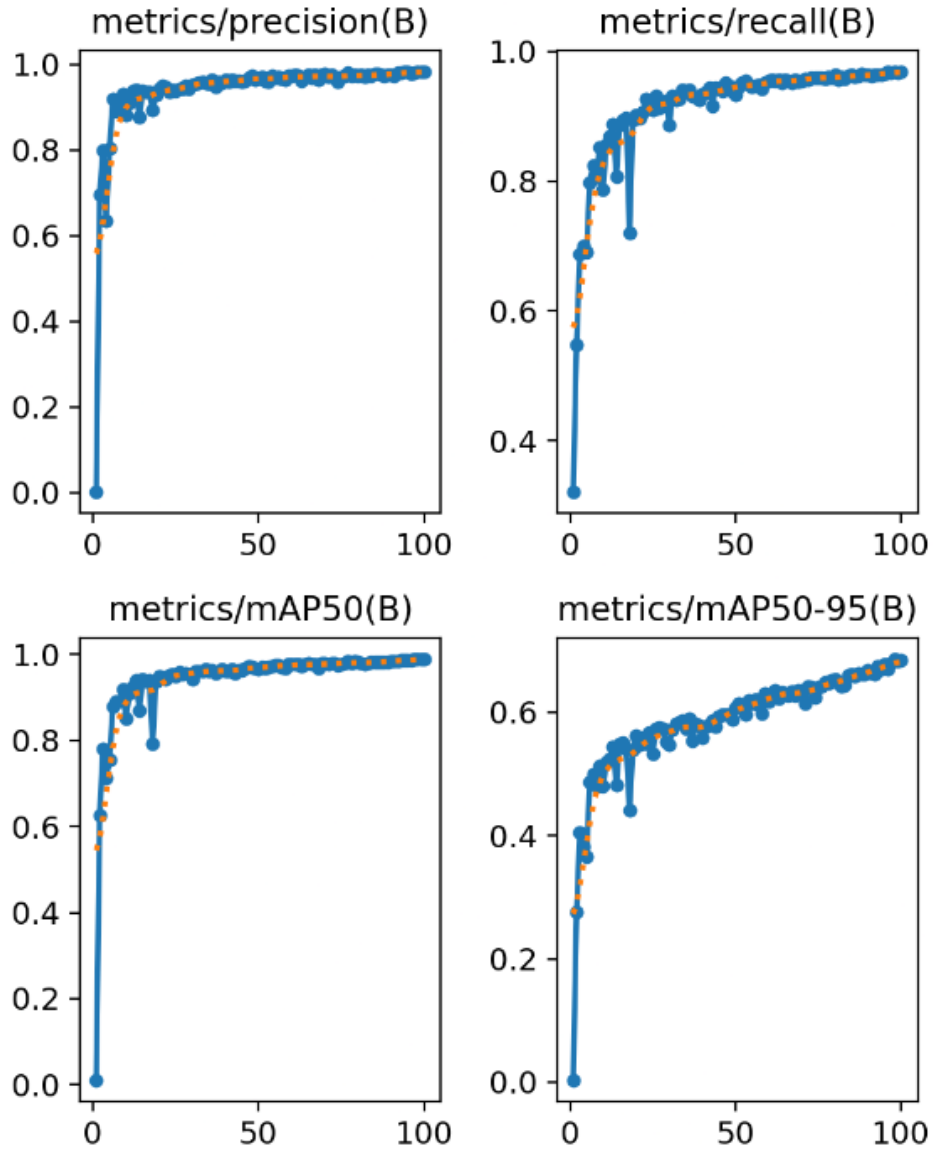


Figure 5.8: Training and Validation Metrics of YOLOv8 Model

- (f) Val/box_loss: This metric is the same as the train/box_loss metric, but it is calculated on the validation dataset.
- (g) Val/cls_loss: This metric is the same as the train/cls_loss metric, but it is calculated on the validation dataset.
- (h) Val/df_l_loss: This metric is the same as the train/df_l_loss metric, but it is calculated on the validation dataset.
- (i) Metrics/mAP50(B): This metric measures the mean average precision (mAP) of the model at an IoU threshold of 0.5.

- (j) Metrics/mAP50-95(B): This metric measures the mean average precision (mAP) of the model at IoU thresholds from 0.5 to 0.95.

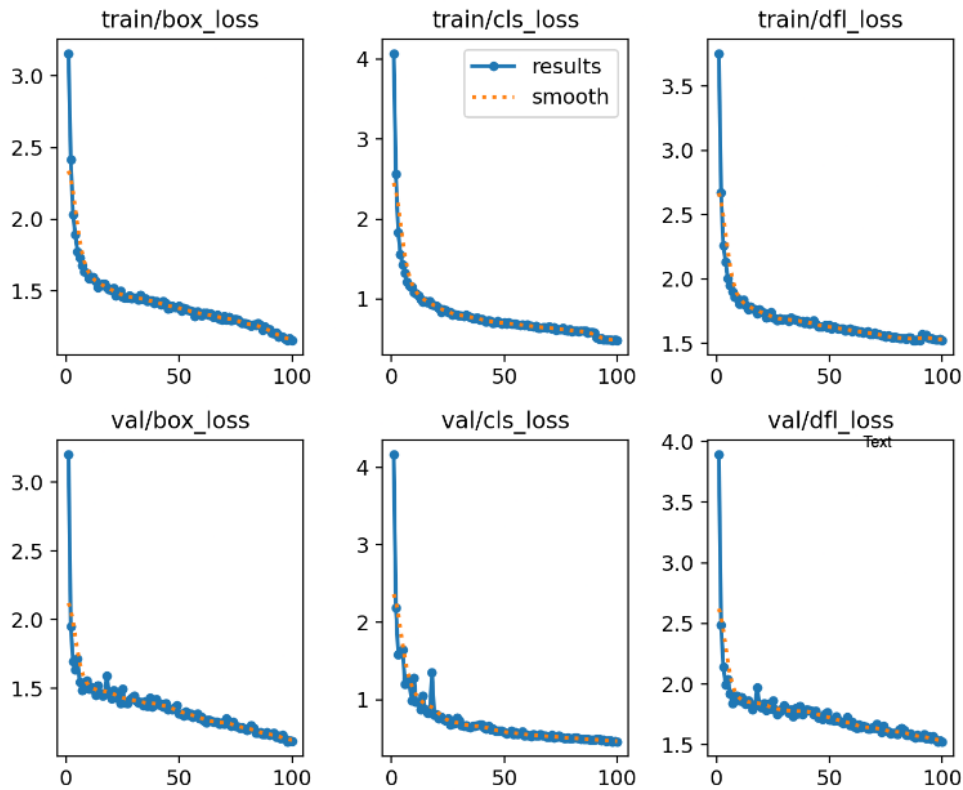


Figure 5.9: Training and Validation Loss of YOLOv8 Model

Hyper Parameters: With YOLO, the version-specific hyperparameter is 100 epochs, patience is 60, and batch size is 16. 0.01% is the initial learning rate. The ultimate rate of learning is only 0.01. A validation box loss of 0.45994 corresponds to the model's ability to find inside the bounding box, while a class loss of 1.5272 indicates the accuracy of the classification. The class imbalance is handled by 0.0000596 dfl loss.

In Figure 5.10, the confidence curves in the image show that the object detection model achieves a precision of 0.817 at a recall of 0.98 and f1 0.95. This means that the model successfully captures a substantial number of genuine positive cases, as shown by the recall confidence of 0.98 and the F1 confidence level of 0.95, which indicates that memory and accuracy have been well-balanced.

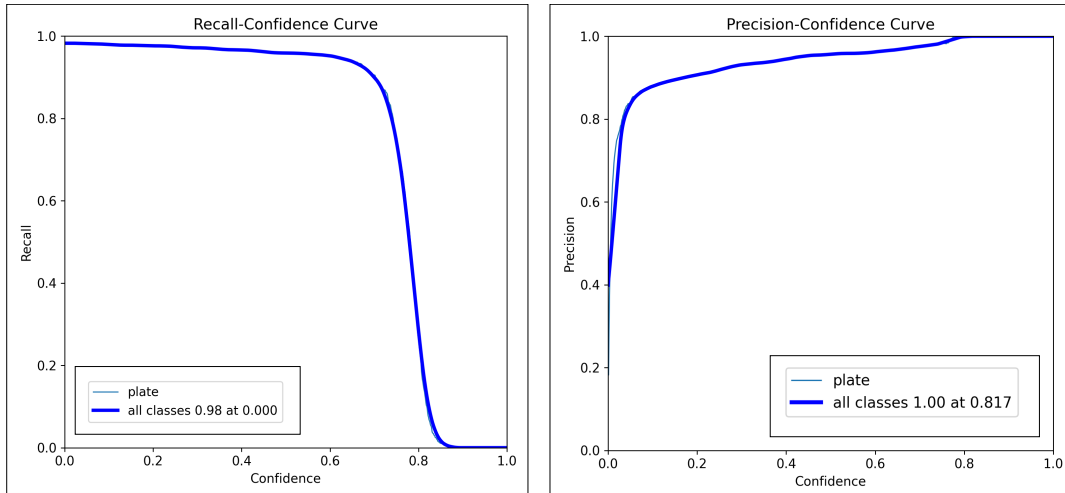


Figure 5.10: Confidence Curves for YOLOv8

5. **Testing:** Every vehicle in the images has been detected and localized by the model, as seen by the red bounding boxes. After testing out the trained model and seeing how well it does. I evaluated how well the model would do when presented with new data. This impressive result shows that YOLOv8 is an effective instrument for detecting objects in images. Overall, the image

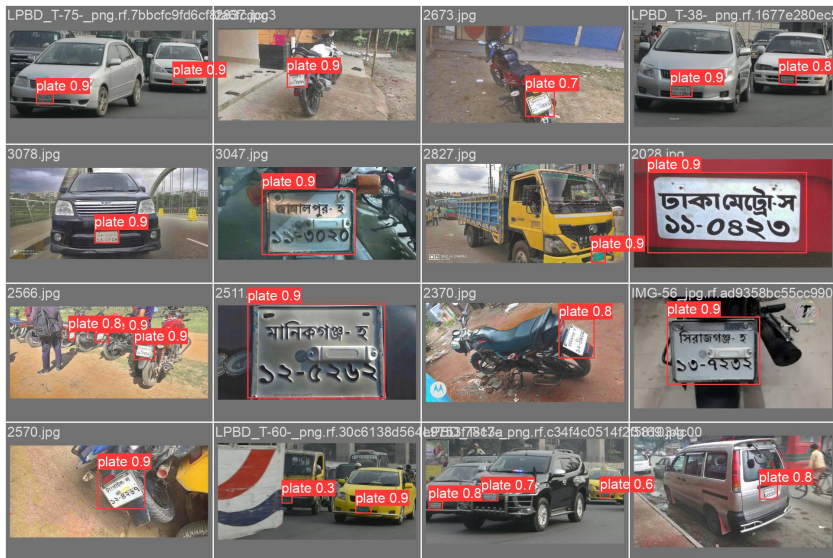


Figure 5.11: Test Result Prediction of Trained Model

demonstrates the superior performance of the YOLOv8 object identification model in identifying and localizing vehicles in images. If we look at figure 5.11 we can see some pictures that are correctly identified by YOLOv8.

According to Figure 5.12, the precision-recall curve in the image shows that the YOLOv8 model achieves a precision of 0.977 at a recall of 0.977. This means that 97.7% of the detections made by the model are relevant, and 97.7% of the ground truth objects in the image are detected.

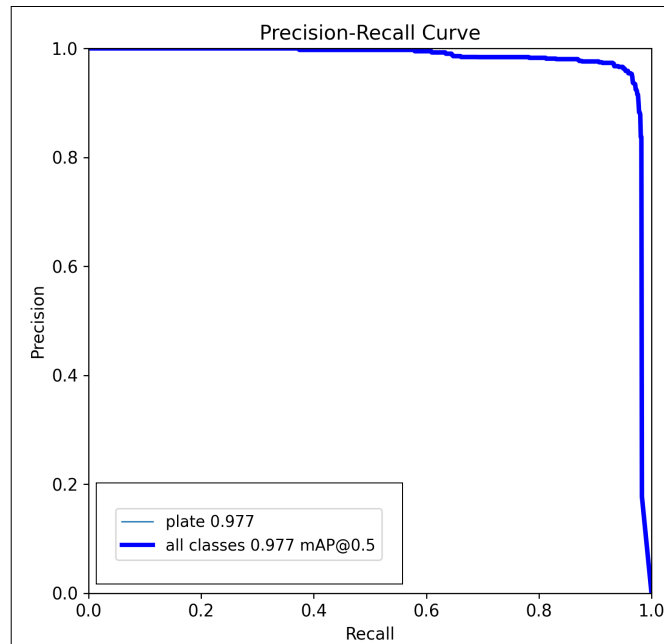


Figure 5.12: Precision-Recall Curve

5.5 Recognition Using OCR

In our study, EasyOCR utilizes a systematic approach specifically developed to accurately analyze and extract textual information written on the Bangla LPs. Here, we will thoroughly comprehend EasyOCR’s language identification process in figure 5.13, especially in relation to Bangla, as it applies to LPs.

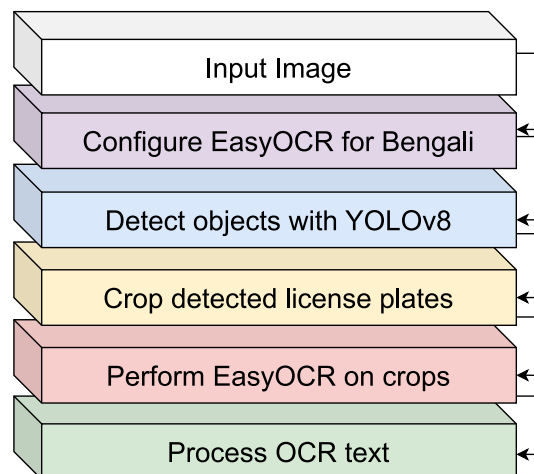


Figure 5.13: Easy-OCR Workflow

5.5.1 Setting Up OCR Reader

Configuring an OCR reader using EasyOCR involves a simple procedure to enable the effective extraction of text from images. To begin, EasyOCR must be installed using the pip package manager to make the library available in the Python environment. After installation, the EasyOCR module needs to be included in the

notebook. First, we initialized the OCR reader by creating an object of the EasyOCR Reader class. Then, specify our desired language, “Bangla,” by supplying a list as a parameter, enabling the reader to identify text in that language. Once the OCR reader is initialized, we load the images we want to analyze. Used the “read-text” method of the reader object to extract text from the pictures. The findings will include data such as the identified text, coordinates of the bounding box, and scores indicating the level of confidence.

5.5.2 Object Detection with YOLOv8

Utilize the YOLOv8 model to identify objects within the selected images by importing them into OCR. For this, firstly, the number plate will be classified from the images. Then, the number plate will be detected by using a bounding box to tell where the number plate is, and that particular object will be specified as the number plate. So, in figure 5.14, we can see some car plates detected by YOLOv8.



Figure 5.14: Object Detection With YOLOv8

5.5.3 Image Cropping (If Objects Detected)

If the object is detected by using YOLOv8, then the region of interest (ROI) will be cropped in order to encompass the LP so that EasyOCR can be used to perform OCR on the cropped image. In figure 5.15 we can see a sample of a cropped image.



Figure 5.15: Cropped Image

5.5.4 OCR on Cropped Images

After applying YOLOv8 and identifying the cropped images of the LPs, we need to apply OCR on the cropped images. Once EasyOCR is installed and the OCR reader is initialized with the proper language settings, loading the cropped picture is a simple process shown in figure 5.16. This may be done by using a PIL picture object or specifying the file path.



Figure 5.16: OCR on Cropped Image

5.5.5 Processing OCR Results

Once the OCR engine has extracted text from an image, the next stages include retrieving the recognized text, employing bounding box information if it is provided, and assessing confidence scores to measure the accuracy of the results. We need to apply the read text technique to the cropped images for effectively extracting text, providing results in the form of recognized text, bounding box coordinates, and confidence scores. It keeps a tally of total confidence scores (total) and recognized character count for calculating an average confidence score.

5.5.6 Calculating Average Confidence Score

The process includes tracking total confidence scores and character counts for the subsequent calculation of an average confidence score, computed by dividing the cumulative confidence score by the count of recognized characters.

Models	Average confidence dataset-1	Average confidence dataset-2
YOLOv5n	0.6497	0.3398
YOLOv5m	0.6648	0.3546
YOLOv8n	0.6551	0.3402
YOLOv8m	0.6843	0.3402

Table 5.1: EasyOCR Average Confidence Score in Different Models for Both Datasets



Figure 5.17: OCR Recognition: True vs. False

EasyOCR is able to identify characters in 2237 out of a total of 2754 photos (81.23%). EasyOCR demonstrates a favorable ability to accurately detect most individual characters; however, when considered collectively, its performance ranges from lower to moderate proficiency. As illustrated in Figure 5.17, instances of both true positives and false positives in EasyOCR recognition are evident. Notably, an observed advantage is that an increase in image enhancement through the application of DCP corresponds to improved OCR capabilities.

Chapter 6

Experimental Evaluation

In this part, we will detail the specifications of the laptop or desktop computer that ran all of our algorithms. Following that, we will demonstrate the superiority of our model over competing models by comparing qualitative and quantitative measurements.

6.1 Experimental Setup

For this study, we have used a 2020 MacBook Air equipped with Apple’s M1 processor, which has eight cores for processing power and seven for graphics processing unit. In order to create fog, we have used the monocular depth method and dcp for dehazing on an Apple MacBook Air (2020). In contrast, a different device with a Gt1030 GPU and an I510400F CPU has been used to do object detection using YOLOv8, and to extract recognition characters using EasyOCR. For the whole study, we have used python 3.9.7.

6.1.1 Basic libraries for Our Model

The latest state-of-the-art YOLO model, YOLOv8, is applicable to instance segmentation, object identification, and image classification [59]. We have used YOLOv8 to train and test our model. The main functions that we have used for implementing the model are given below.

YOLO() A constructor method. It is used to create or load a model. We have started a new model with ‘yolov8n.yaml’ parameter.

train() Used to train a model. Passed parameter ‘data’ holds the location of a .yaml that holds the train dataset location; passed parameter ‘epoch’ holds the number of times the model will be trained with the same dataset.

predict() Used to predict the location of a target in an image. Takes the location of an image or an image instance as a parameter. Returns the results of the prediction.

val() Used to validate the trained model.

6.1.2 Basic libraries for OCR

We have used EasyOCR to run optical character recognition on our detected LP.

Reader() A constructor method that takes a list of languages as parameters and returns an instance capable of detecting text of given languages.

readtext() It takes an image as a parameter and returns the text, text position, and confidence score.

6.2 Experimental Findings

The findings of our study will be presented in this segment. After establishing that the artificially added fog is not irrelevant, we compare our suggested model with other standard models and alternative versions.

6.2.1 Customised Dataset’s Findings

The monocular depth in the image was determined using beta values of two, three, and four. In order to induce fog in the image, we have compared it to an actual foggy image and rendered the ground truth foggy to determine whether or not its intensity matched that of the real fog. Figure 6.1 illustrates how a foggy image is produced from two images of the ground truth—real fog and monocular depth. After the fog subsides, the appearance is comparable to that which results from comparing natural and artificial fog. There are several minor drawbacks, such as the lack of foreground fogginess to create depth of field, which is possible with monocular depth. Upon examining the intensity graphs of both images in figure 6.2, it becomes evident that the artificial fog curve and the actual fog curve are parallel or identical. That indicates the fogginess is nearly adequate, with the exception of the density.

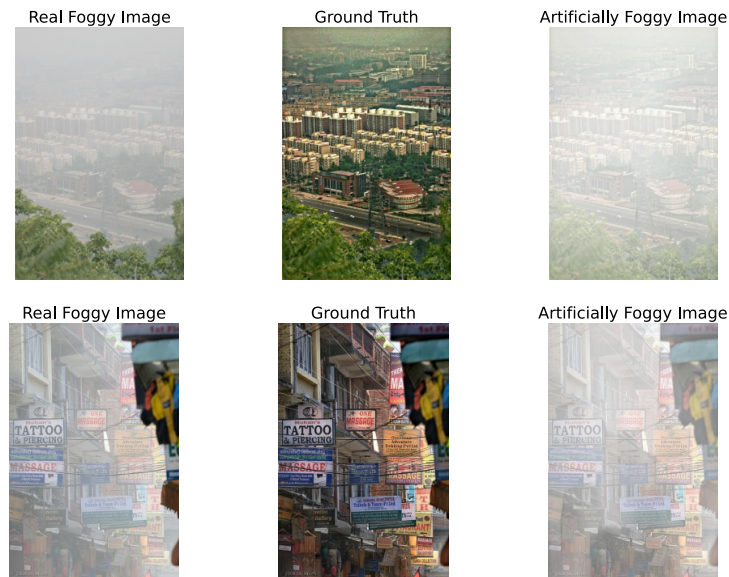


Figure 6.1: Visual Comparison of Ground Truth, Real Foggy Images and Monoculardepth Foggy Images

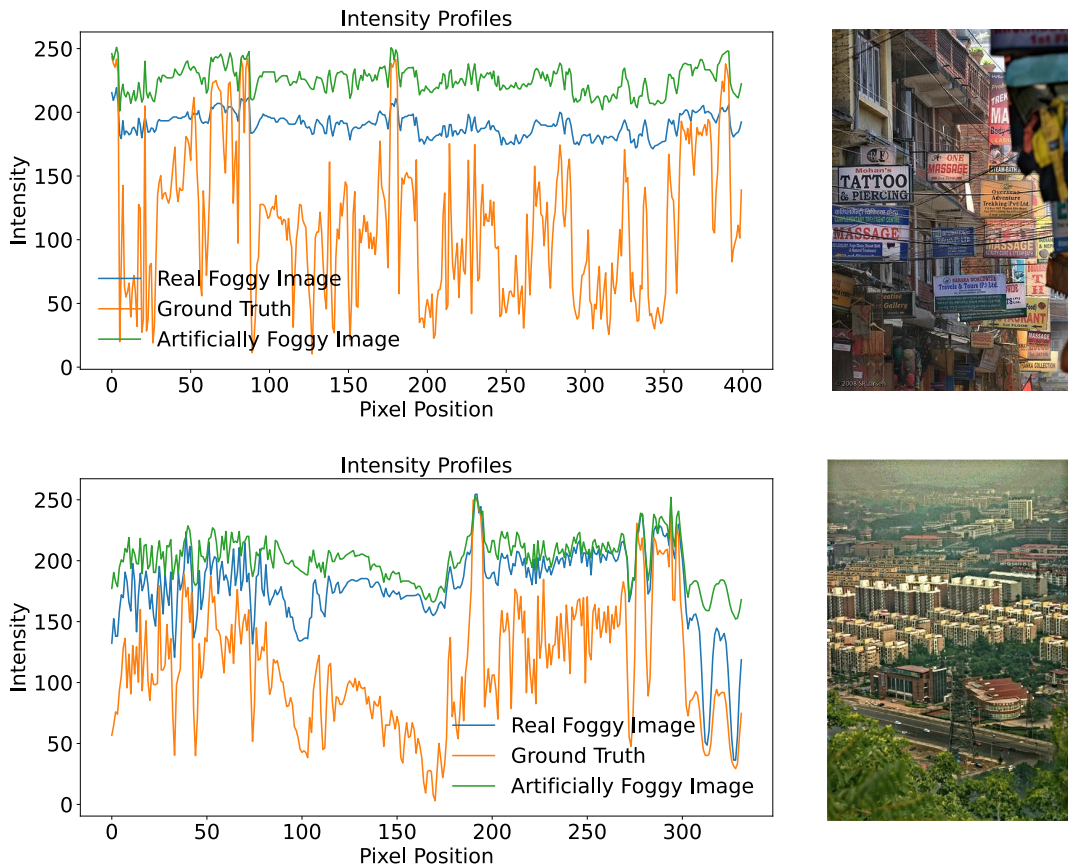


Figure 6.2: Frequency Intensity of Ground Truth, Real Foggy Images and Monoculardepth Foggy Images

6.2.2 Qualitative Evaluation

Our customized dataset shows ten real-world photos in Fig. 6.3. Actual foggy images are depicted in column (a), while the results from various defogging methods are shown in columns (b) to (h): (b) He et al.[3], (c) Berman et al.[14], (d) Xie et al. [5], (e) Kumari et al. [43], (f) Meng et al. [9], and (g) Cai et al. [15]. Columns (h) display the outcome of the proposed method. Figure 6.3 shows that algorithms (d) and (e) produce distorted colors in the recovered picture, especially in the car area; techniques (b) and (f) have visible distortions in areas with varying depth values.

We attempt to compare the transmission map in color in figure 6.4. On the cool-warm colormap transmission map, the multi-scale retinex-based algorithm might have seemed to have the greatest visibility. In the given diagram, a positive value, denoted by the color red, is greater than the central value. Conversely, a negative value or one that is lower than the central value is represented by the color blue [65]. It is evident that the MSR is oversaturated and pale, indicating that most of the color is situated near the median. This indicates that the image’s luminosity or light is excessive, rendering the region difficult to differentiate due to overexposure. Next, if we observe fast restoration, the swan and forest picture portion will become darker blue, indicating that its depth has been precisely measured. However, the crimson portion of the image contains fewer hues than the remainder. Our hybrid algorithm resembles the actual DCP, except that it calculates depth more precisely and contains more hues. The illumination is achieved through well-balanced color

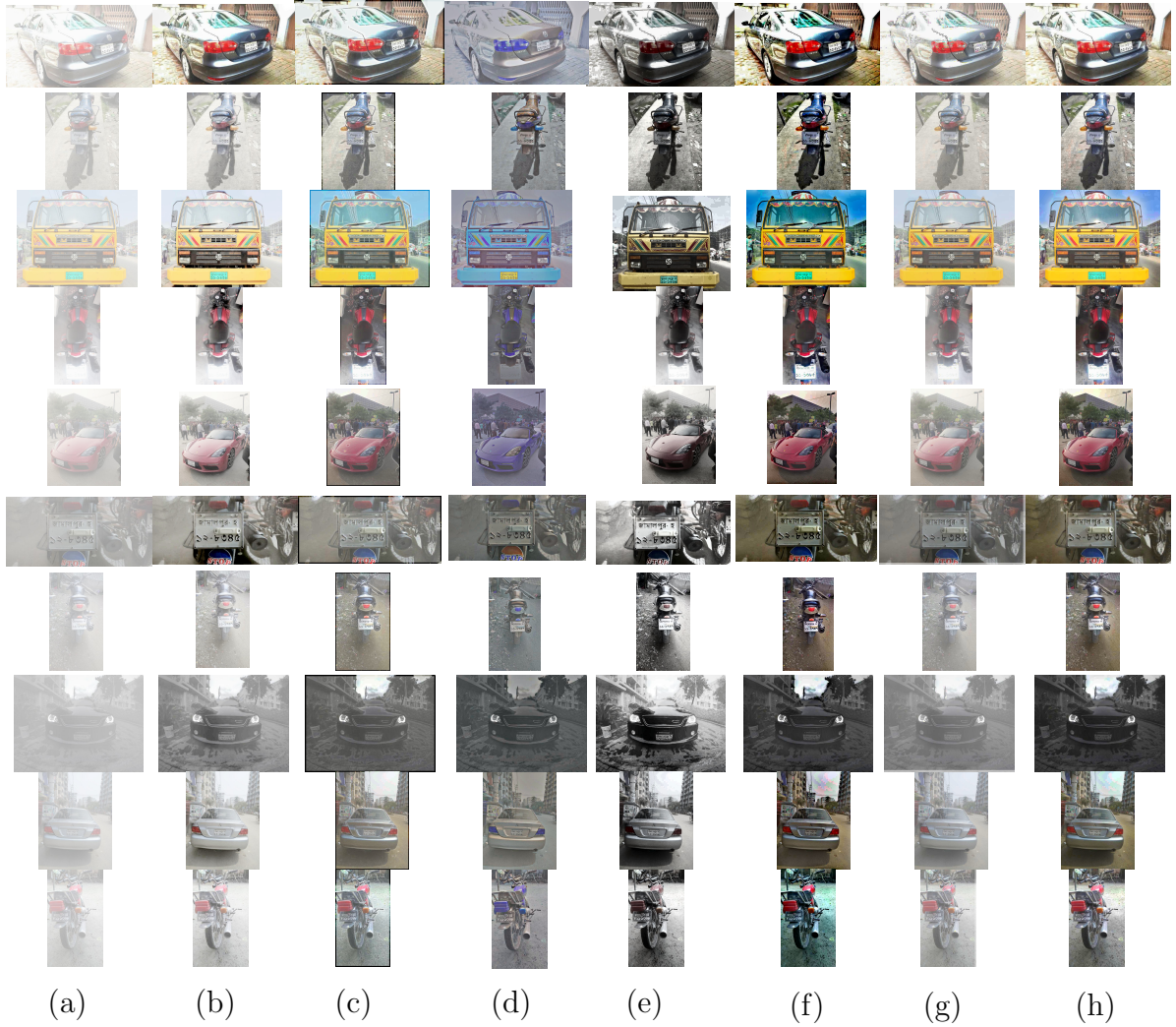


Figure 6.3: Performing a Qualitative Analysis of 10 Images From Our Customized Unique Dataset Using Several Algorithms (a) Hazy Images With Fog, (b) He et al.[3], (c) Berman et al.[14], (d) Xie et al. Method, (e) Kumari et al. Method, (f) Meng et al. Method, and (g) Cai et al. Method, (h) Proposed Method.

saturation and the greater the number of tones, the more specific each region is.

We compare three samples using the customized dataset to see where our proposed model and He et al. perform significantly better than the others shown in figure 6.5, where both allow for more vibrant colors. While Cai et al. and Xie et al. are easy to notice, one color is too saturated, and another is not vibrant at all. He et al. are far from the initial ground truth. Perhaps leading to deceptive information inconsistency [61] there may be information loss and problems in hazardous conditions because of the fading sample's lower intensity values compared to others.

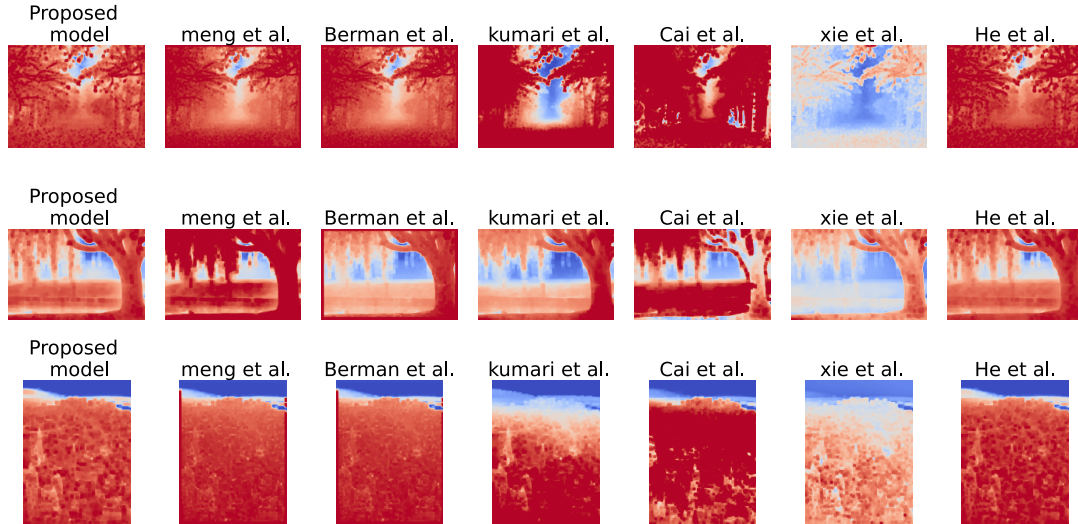


Figure 6.4: Comparative Visualization of Transmission Maps for Three Sample Images Processed by Dehazing Algorithms



Figure 6.5: Comparative Analysis of Seven Image Processing Algorithms on Three Images From a Custom Dataset, Showcasing the Original Alongside Each Algorithm's Output.

6.2.3 Quantitative Evaluation

Table 6.1 and table 6.2 show the appropriate index values that were obtained by computing the SSIM index over the restored pictures of dataset-1 and dataset-2 displayed in figure 6.6 and figure 6.7.

The SSIM is a technique used to quantify the degree of similarity between two images. It is computed on several windows of an image. The distance between two windows is shown in the equation 6.1, where x , and y , with equal dimensions $N \times N$

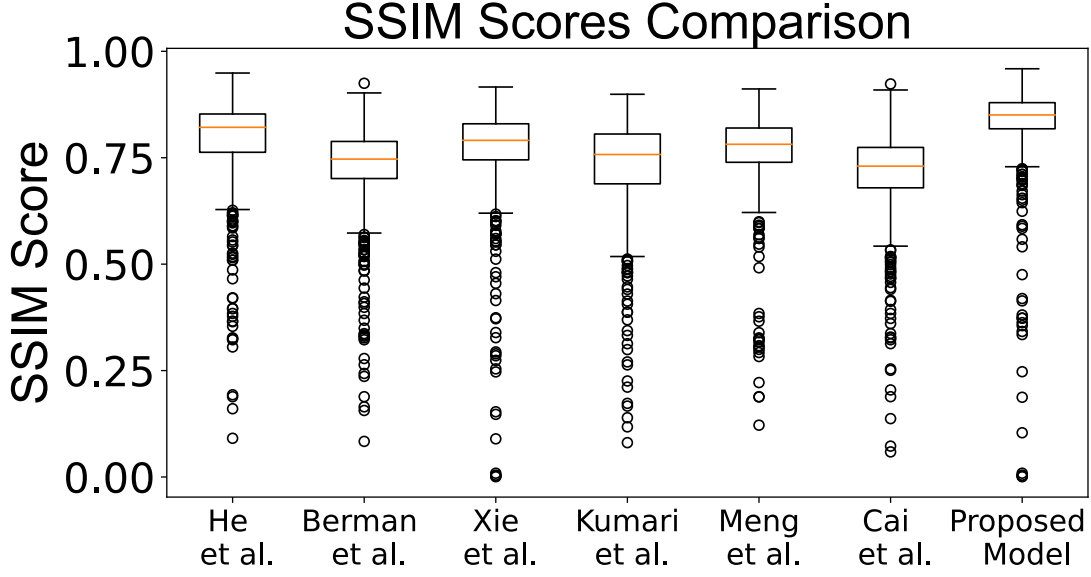


Figure 6.6: SSIM Graph for Dataset-1

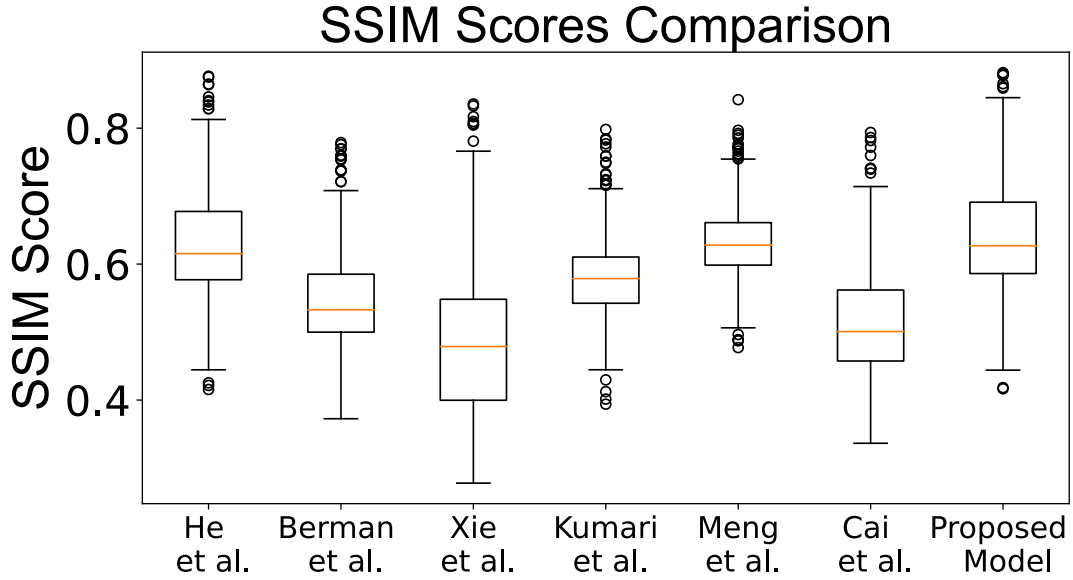


Figure 6.7: SSIM Graph for Dataset-2

[10].

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (6.1)$$

Here, μ_x, μ_y are the average of x, y . σ^2 is the variance and σ is covariance of x, y . $C = (k.L)^2$ where C is the factors that stabilize the division when the denominator is weak. Besides, L represents the range of possible pixel values and $k=0.01$ and 0.03 by default. The result of this formula might range from -1 to 1 , with 1 being the highest degree of resemblance [4].

By comparing our findings to those of the seven other techniques, we find that our suggested model outperforms them all in both datasets. Our suggested approach is a state-of-the-art method, in contrast to the other techniques, which display lower

Vehicle ID	He et al.	Berman et al.	Xie et al.	Kumari et al.	Meng et al.	Cai et al.	Proposed Model
V1575	0.841	0.789	0.72	0.684	0.719	0.715	0.859
V54	0.791	0.767	0.839	0.695	0.629	0.77	0.839
V394	0.886	0.831	0.845	0.719	0.827	0.855	0.883
V384	0.86	0.786	0.802	0.845	0.795	0.75	0.873
V44	0.822	0.729	0.775	0.799	0.801	0.701	0.838
V813	0.773	0.549	0.69	0.528	0.763	0.643	0.773
V803	0.869	0.821	0.889	0.679	0.796	0.81	0.901
V1290	0.904	0.721	0.836	0.546	0.883	0.621	0.912
V1514	0.821	0.737	0.777	0.788	0.799	0.714	0.833
V133	0.796	0.698	0.787	0.812	0.718	0.667	0.818
Average	0.8363	0.7428	0.796	0.7095	0.773	0.7246	0.8529

Table 6.1: SSIM Values of Different Algorithm on Dataset-1

Vehicle ID	He et al.	Berman et al.	Xie et al.	Kumari et al.	Meng et al.	Cai et al.	Proposed Model
7_jpg.	0.610	0.527	0.526	0.603	0.583	0.497	0.623
IMG-44	0.714	0.635	0.510	0.600	0.773	0.588	0.747
28_jpg	0.601	0.528	0.441	0.532	0.570	0.524	0.589
126-jp	0.682	0.597	0.590	0.635	0.617	0.605	0.692
90_jpg	0.711	0.643	0.677	0.635	0.672	0.614	0.737
IMG-1_	0.579	0.508	0.497	0.526	0.575	0.475	0.591
80_jpg	0.643	0.567	0.407	0.595	0.638	0.543	0.639
152-jp	0.682	0.553	0.436	0.610	0.653	0.559	0.700
IMG-11	0.654	0.558	0.571	0.596	0.655	0.544	0.672
IMG-18	0.506	0.429	0.290	0.546	0.591	0.399	0.518
Average	0.638	0.554	0.494	0.587	0.632	0.534	0.650

Table 6.2: SSIM Values of Different Algorithm on Dataset-2

SSIM index values. The findings of the SSIM index seem to align well with the visual judgment shown in table 6.1 and 6.2.

In the SSIM box plot in figure 6.6 for dataset-1 and 6.7 for dataset-2, we can see that our suggested model has the highest median of 0.8301 for dataset-1 and for dataset-2, the highest median we get from two models one is Meng et al. and the other is our proposed model, both have a similar median value of 0.627. The median is represented by the orange line in this box. In the case where our proposed methods achieved the greatest median, Cai et al.’s model produced the lowest outcomes (median=0.7302) for dataset-1, and for dataset-2, the lowest outcome was 0.478, which is from Xie et al.’s model. Each distribution has an upper and lower quartile; the upper is located midway between the median and the maximum value, while the lower is midway between the median and the lowest possible value. Our suggested model indicates that most values lie close to the median for dataset-1 and for dataset-2, both our suggested model and Meng et al.’s model.

PSNR evaluates how well a compressed video or picture can be reconstructed. To determine it, we take the MSE from both the original and compressed images [16].

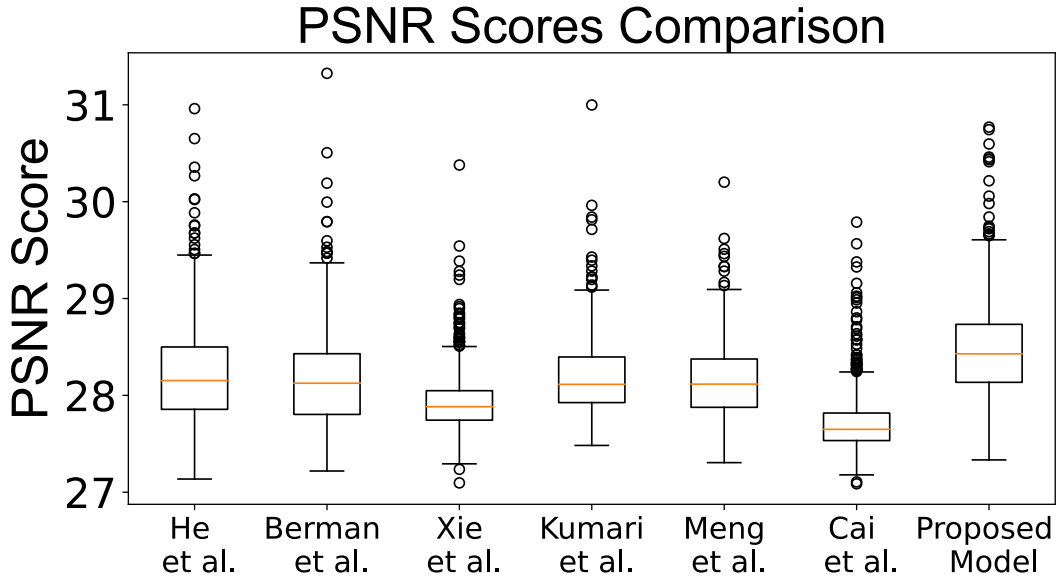


Figure 6.8: PSNR Graph for Dataset-1

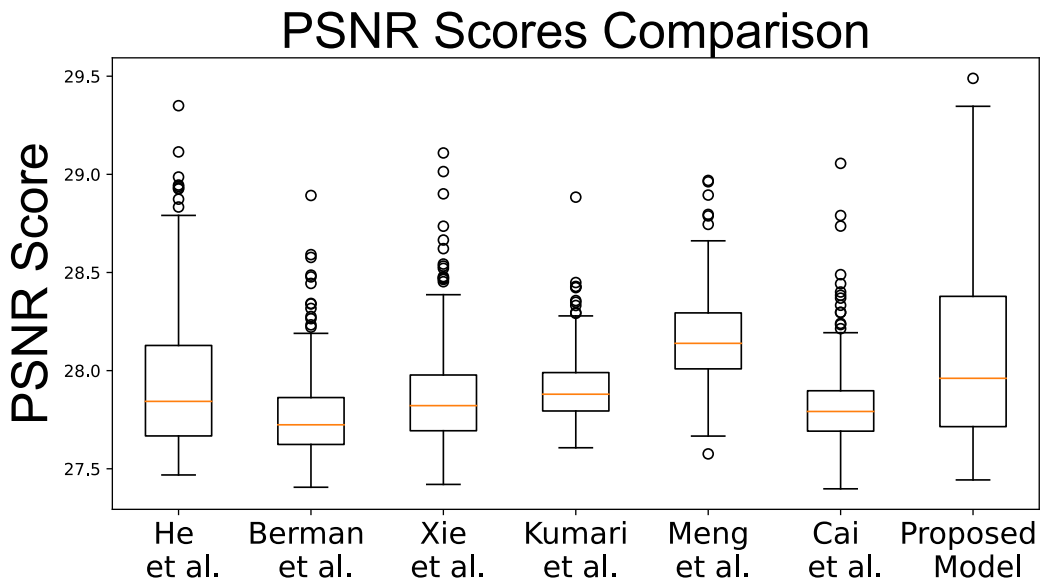


Figure 6.9: PSNR Graph for Dataset-2

For PSNR, the formula is 6.2.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \quad (6.2)$$

Now, for better understanding, we took seven hundred-plus images from our custom dataset and ran different algorithms for PSNR. In an 8-bit picture, MAX_I is the highest possible pixel value, while MSE is the mean squared error in a compressed image. Decibels (dB) are the usual units of measurement for PSNR. The picture or video quality after compression improves as the PSNR rises.

Data quality diverges within a specific range, as illustrated by the PSNR box plot in figure 6.8 for dataset-1 and in figure 6.9 for dataset-2, with additional data scattered

in lower and higher whiskers. By observing the extensive range of the box and whiskers in the PSNR box plot, we may deduce that the proposed model possesses a widely dispersed PSNR. Our suggested approach has a median value of 28.429 (dataset-1), which is the highest among all and the second highest out of all the algorithms for dataset-2, is 27.960.

In Table 6.3, the PSNR analyses using our ten customized pictures are shown for dataset-1 and in Table 6.4 for dataset-2. The findings showed that the suggested method is much better than the others for dataset-1 and for dataset-2, Meng et al.'s model shows better performance; the only other way with a comparable PSNR value is the one described by He et al. [3] for dataset-1, and for dataset-2 there is no comparable model.

By comparing the suggested technique to state-of-the-art methods in terms of PSNR value and SSIM index for dataset-1, the experimental findings show that the former produces comparable or superior outcomes.

ID	He et al.	Berman et al.	Xie et al.	Kumari et al.	Meng et al.	Cai et al.	Proposed Model
v1575	29.525	28.006	27.709	28.384	28.548	28.684	29.722
v54	28.474	28.691	27.849	27.942	27.726	27.819	28.497
v394	27.964	28.41	28.036	27.991	28.42	27.834	28.099
v384	28.652	27.867	27.814	28.556	28.507	28.048	28.573
v44	28.149	28.295	27.734	28.356	28.092	27.685	28.313
v813	27.572	27.475	27.659	27.887	28.036	27.692	27.4935
v803	28.8	27.967	28.601	27.811	27.726	27.451	29.084
v1290	30.053	27.534	27.701	28.609	28.789	27.666	30.459
v1514	28.334	27.869	27.939	27.918	28.308	27.767	28.591
v133	28.296	28.132	27.991	28.495	28.025	27.603	28.43
Average	28.5819	28.0246	27.9033	28.1949	28.2177	27.8249	28.72615

Table 6.3: PSNR Values of Different Algorithm on Dataset-1

Vehicle ID	He et al.	Berman et al.	Xie et al.	Kumari et al.	Meng et al.	Cai et al.	Proposed Model
7_jpg.	28.022	27.922	27.880	28.063	28.083	27.829	28.378
IMG-44	27.788	27.742	27.783	27.787	28.046	27.731	27.710
28_jpg	28.088	27.724	27.657	27.893	27.887	27.930	28.397
126-jp	28.117	28.048	28.271	28.056	27.995	28.173	28.157
90_jpg	27.888	27.599	28.310	27.853	28.151	27.471	27.876
IMG-1_	27.632	27.696	28.189	27.978	27.989	27.783	27.735
80_jpg	28.366	28.024	27.764	27.877	28.094	27.966	28.562
152-jp	28.142	27.732	28.050	27.729	27.961	27.748	28.217
IMG-11	28.730	27.946	27.812	27.781	28.247	27.707	28.711
IMG-18	27.537	27.575	27.883	27.941	28.031	27.729	27.532
Average	28.031	27.801	27.960	27.896	28.048	27.807	28.128

Table 6.4: PSNR Values of Different Algorithm on Dataset-2

Using the ten images from our custom dataset, the Naturalness Image Quality Evaluator (NIQE) analyses are shown in Table 6.5. With a NIQE index of 22.552, Xie et al. [5] was found to be much better than the others. Our suggested model was the only other approach with a similar NIQE value. The Naturalness Image Quality Evaluator, or NIQE, takes an image’s statistical characteristics and compares them to a model of how statistics behave in natural images. According to Mittal, a lower NIQE score suggests that the original picture’s features are more accurately captured, which usually means that the image quality is higher [10]. These features are acquired independently of any distorted versions or images judged by humans. When evaluating all approaches, it can be concluded that Xie et al. [5] outperformed state-of-the-art procedures regarding NIQE value. The experimental results confirm this. In contrast, our suggested model achieves the lowest minimum index, which is almost similar to Xie et al. [5].

ID	He et al.	Berman et al.	Xie et al.	Kumari et al.	Meng et al.	Cai et al.	Proposed Model
V1575	19.035	17.520	19.971	23.787	25.437	26.280	19.331
V54	22.857	21.779	19.646	21.963	28.194	22.385	20.804
V394	21.649	26.760	22.218	27.467	26.102	25.783	22.501
V384	30.294	31.160	25.267	26.063	24.393	34.047	27.520
V44	23.585	21.238	19.910	22.172	20.329	26.956	19.343
V813	23.992	24.259	21.041	23.184	21.554	25.556	22.410
V803	27.440	27.124	25.385	26.227	26.563	30.170	24.123
V1290	29.662	26.269	24.623	22.591	23.228	32.163	27.656
V1514	25.011	27.339	24.603	20.470	21.293	28.622	21.210
V133	27.332	24.814	22.852	22.514	29.000	27.022	21.825
Average	25.086	24.826	22.552	23.644	24.609	27.898	22.672

Table 6.5: NIQE Values of Different Algorithm on Dataset-1

6.2.4 Findings from YOLOv8

In the experimental phase, five distinct models are employed to identify LPs. The initial iteration of YOLO involved bounding box detection to discern licensed content, with multiple YOLO variants implemented on separate backbones. These YOLO iterations encompass v8n, v8m, v5n, and v5m. Notably, in terms of parameters and floating-point operations per second, YOLOv8m exhibits twice the computational potency of YOLOv8n. For YOLOv8m, we use CSPdarknet50 as the backbone and 100 epochs to train our datasets for detection. For other models, optimal parameters, which may outperform, have been used.

A comprehensive evaluation shown in table 6.6 is conducted on 15% of the dataset images following a predefined training and validation period for both dataset-1 and dataset-2. Results indicate that YOLOv8m achieves the highest accuracy at 0.985 for dataset-1 and 0.80 for dataset-2, a commendable F1 score of 0.96 for dataset-1 and 0.78 for dataset-2, a recall confidence of 0.98 for dataset-1 and 0.80 for dataset-2, and precision at 1.00 for both dataset. Specifically, YOLOv8m demonstrates an accuracy of 0.985 and a recall rate of 0.98 at a confidence threshold of 0.430 for dataset-

1 and for dataset-2 accuracy, and the recall rate is 0.80 at a confidence threshold of 0.310 for dataset-2. surpassing the performance of YOLOv5n and YOLOv5m, which achieved 0.971 and 0.982, respectively, and YOLOv8n with a score of 0.942 for dataset-1. For dataset-2, the performance of YOLOv5n, YOLOv5m, and YOLOv8n achieved 0.79, 0.68, 0.68. The hyperparameters for YOLO8m are patience = 50, batch size = 16, learning rate initial = 0.01, and learning rate final = 0.01. Since we have an adequate amount of samples and our main contribution is in the form of data pre-processing, we skip the use of augmentation.

Detection Model	Dataset-1 Accuracy	Dataset-2 Accuracy
YOLOv8m	0.98	0.80
YOLOv8n	0.94	0.68
YOLOv5m	0.98	0.68
YOLOv5n	0.97	0.79

Table 6.6: Accuracy Evaluation of YOLO Models Across Two Datasets

6.2.5 Before Applying DCP and After Applying DCP

The significance of these efforts may be better understood by comparing the two precision-recall graphs in figure 6.10: one shows the detection result of using DCP in hazy images, while the other shows the result without applying DCP. Before adding DCP, the precision-recall curve in the picture demonstrates that the YOLOv8 model achieves a precision of 0.623 at a recall of 0.623. This equates to 60% accurate model detections, which means 60% ground truth object detections in the picture. Contrarily, the image’s precision-recall curve reveals that the YOLOv8 model attains a recall of 0.962 after DCP is employed. This indicates that almost all of the ground truth items in the picture were discovered, with a 98.5% success rate for the model’s detections.

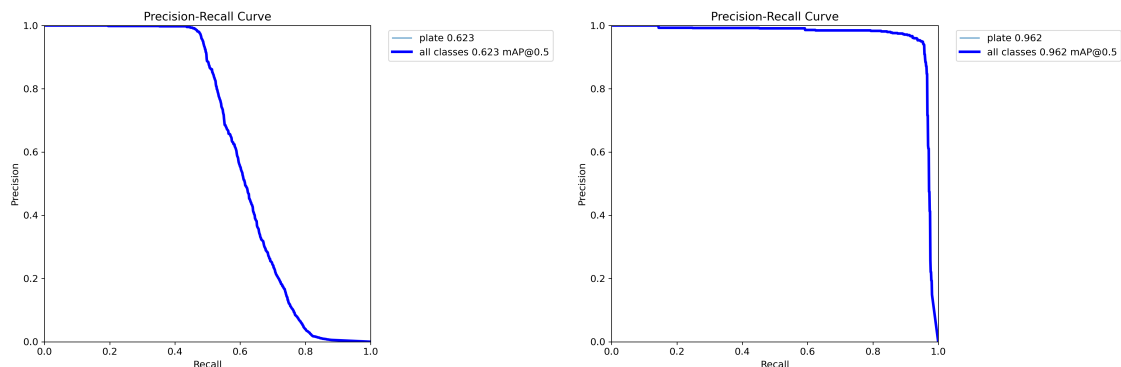


Figure 6.10: Before applying DCP and after applying DCP

6.2.6 Performance Evaluation of EasyOCR

The trained model can recognize a LP in a picture and provide its approximate position and size inside a bounding box. We have then cropped the image according to the bounding box and have given the cropped image to EasyOCR to detect the text inside the LP area. We have cropped 100 images and ran EasyOCR on them. EasyOCR could tell that 98 of the images had text in them. Although all the texts that OCR has detected are not 100% accurate, the OCR has detected more text on LPs after DCP has been applied to images. The increased contrast on DCP-applied images helped in the OCR process by a great margin. However, the accuracy has been only at 47% with an average of 68.43% confidence from EasyOCR. Unfortunately, only high-resolution images with perpendicular LPs with proper fonts have been detected appropriately. Also, the system has not been able to provide the license with the appropriate sequence. Sequences like 6.11 are very

['হ', 'কিশোরগঞ্জ-', '২-৪০২৮']

Figure 6.11: Wrong Sequence LP

common. Also, damages in LPs, handwritten LPs, and tilted LPs have very low accuracy and confidence.

6.2.7 Overview of Findings

After applying all of the methods, if we look at figure 6.7, we can observe that when we use Monodepth to add fog, the average absolute error is close to 12%, and when we use DCP to dehaze the picture, the average SSIM is almost 0.8301 for dataset-1 and 0.627 for dataset-2. Beside, the average PSNR is 28.429 for dataset-1 and 27.96 for dataset-2. Our important goal is to remove haze from a picture so that we can effectively recognize the number plate using the YOLOv8 algorithm. After applying YOLOv8, we are able to detect almost all of the number plates with 98.5% accuracy for dataset-1 and 80% for dataset-2. Then, to recognize the characters, we use EasyOCR, and the algorithm’s average confidence score is 68.43% for dataset-1 and 34.02% for dataset-2.

Methodology	Parameters		Value
Monoculardepth	Average Absolute Error		0.122
Dark Channel prior	Average SSIM	Dataset-1	0.8301
		Dataset-2	0.627
	Average PSNR	Dataset-1	28.429
		Dataset-2	27.96
YOLOv8m	Accuracy	Dataset-1	98.50%
		Dataset-2	80%
EasyOCR	Average Confidence	Dataset-1	0.6843
		Dataset-2	0.3402

Table 6.7: Parameters Overview

Chapter 7

Limitations and Future Work

Using the DCP fog-dehazing method, this study gets significant accuracy in pin-pointing and identifying Bangla car plates in hazy environments. The reason for choosing Bangla as the target language is that it is considered a low-resource language due to its high digital text complexity and the few accessible resources. We used DCP for dehazing images. We have also presented an algorithm that outperforms other algorithms; nevertheless, there are a few drawbacks that need to be addressed. In this part, we will discuss these issues along with some ideas for further study.

7.1 Limitations

In our study, we have proposed a DCP-based method to tackle the challenge of identifying and detecting Bangla car plates using YOLOv8 and EasyOCR. After deeply analyzing the DCP-based defogging algorithm, this study first points out some of its main drawbacks.

1. **Inadequate image resolution:** DCP is primarily concerned with haze removal and may not fully resolve color shifts caused by haze. This causes a loss of color integrity in the dehazed photos, affecting the overall visual quality. As a result, including the DCP method resulted in a substantial decrease in image quality. This deterioration hampered the effectiveness of our OCR technology, limiting its capacity to recognize the actual text on LPs. To overcome this constraint, it is necessary to do more study on other algorithms or make improvements to the DCP algorithm to maintain picture quality while performing the recognition process.
2. **Absence of Real-World dataset:** A significant limitation is our dataset's lack of authentic foggy images. We have constructed our dataset utilizing a monocular depth estimation approach, which may not comprehensively encompass the intricacies and fluctuations in real foggy environments. Real-world fog provides an extensive range of characteristics, such as fluctuating density, thickness, and visibility conditions that are difficult to recreate adequately in synthetic datasets. As a result, the model's performance in the study may not completely represent its effectiveness in dealing with the diverse and unpredictable foggy conditions encountered in practical situations. The lack of

real-world hazy images in the testing dataset may restrict the external validity of the results and impair the model’s performance when applied to real-world circumstances.

3. **Color matching between the car and the plates:** There is an additional constraint that results from difficulties in precisely detecting vehicle LPs and effectively recognizing the texts they contain. This challenge is most noticeable in cases when there are variations in the alignment of LPs. When the color of a vehicle perfectly matches the color of its LP or the characters, YOLO faces considerable challenges in recognizing the car plate. Consequently, OCR has difficulty precisely interpreting the numbers on the car’s LP in such situations. Furthermore, the vertical positioning of LPs poses an additional challenge. The arrangement and alignment of the plates might provide challenges for detection algorithms, hence adding complexity to the process of precisely identifying and comprehending the information on the vehicle plates.
4. **Attachment of obstructions to the number plate:** The presence of obstructions attached to the number plate poses a significant challenge for automated recognition systems. The presence of foreign materials or impediments on the number plate surface, such as dirt, stickers, or other components, inhibits the correct recognition and interpretation of the plate’s information. These obstacles might hamper the visual clarity of the characters, resulting in possible errors in recognition.
5. **The presence of the BRTA seal:** Another limitation is, because of the presence of the BRTA seal and other objects it has become challenging to identify characters. In the figure 7.1 the characters are shown in the red box. But the screw and the seal of BRTA are trash items. As the trash items are up to a significant size, we are facing difficulties while detecting objects by using YOLOv8. For this, we are getting error in outputs for the detection of LPs.

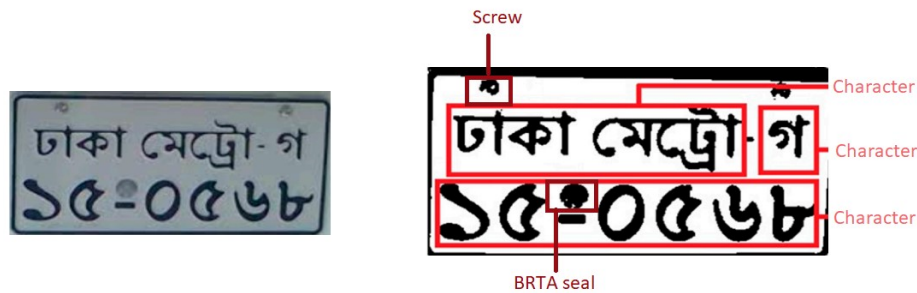


Figure 7.1: Difficult to Identify Characters [37]

7.2 Future Work

Our study has shown the effectiveness of our suggested DCP-based approach for identifying car plates in the Bengali Language using YOLOv8 and EasyOCR. Nevertheless, several constraints and limitations have been recognized, which will serve as a foundation for future study and improvements.

1. **Enhancement of the DCP Algorithm:** The DCP approach, while successfully eliminating haze, has limitations in preserving color accuracy, affecting image quality and OCR performance. Subsequent studies will focus on investigating other dehazing techniques or enhancing the DCP algorithm to mitigate haze-induced color distortions. The goal is to improve the picture's quality while maintaining the identification process's efficiency.
2. **New Dataset with Genuine Foggy Images:** In order to enhance the resilience of our model, our next endeavors will concentrate on obtaining or producing a heterogeneous dataset that accurately portrays the intricacies of fog, encompassing fluctuations in density, thickness, and visibility. This will guarantee that the model's performance is a more accurate reflection of realistic circumstances and can adjust to the unexpected characteristics of real-world foggy conditions.
3. **Managing Deviations in LP Alignment:** Further work is required to address the challenges associated with accurately recognizing LPs, particularly in situations when there is color similarity between the plate and the vehicle, as well as differences in plate alignment. Subsequent investigations will go into sophisticated methodologies for managing inconsistencies in LP positioning and color synchronization. This involves improving the YOLO algorithm to enhance its ability to recognize LPs and overcome difficulties presented by vertically oriented plates. In order to resolve the BRTA seal issue, the output of the OCR can be processed to remove all the garbage and keep only the characters. Another approach for resolving can be, while annotating the dataset, the garbage items like screws or the BRTA seal can be labeled as unnecessary items so that it will not raise any problems during character recognition. As a result, the system's accuracy in interpreting and comprehending LP information will be increased.

To enhance the capabilities of our suggested technique and ensure its usefulness in many difficult real-world situations, we want to focus on these topics in future studies. These advances will lead to the creation of more resilient and reliable systems for Bangla car plate identification under varied environmental circumstances.

Chapter 8

Discussion

Using the Hybrid DCP dehazing methodology combined with the sophisticated YOLOv8 object identification model, we have provided a new method to ALPD that is optimized for difficult foggy environments. While there has been some work on ALPD, it has not been successful in detecting LPs under hazy or foggy conditions. However, eliminating the haze prior to LP identification was our primary focus during the study. Prior to using YOLOv8 for LP detection, we used Hybrid DCP to dehaze the images. Bangla is a language that is often disregarded while dealing with the complexity of digital texts because of its low resource status; this approach is created to enhance the detection and identification of Bangla automobile plates. A crucial element in guaranteeing the visibility and accuracy of LP identification is the DCP algorithm's ability to successfully reduce fog, which greatly improves image clarity. From the dehazed images, YOLOv8 delivers a very efficient way to recognize Bangla LPs. The remarkable plate recognition accuracy of 98.5% was achieved by this method, which has shown to be very successful for dataset-1. Among the data problems that affect the dataset-2 are the presence of salt-paper noise, and a picture that has been inverted. Through the use of the Gaussian Filter, we have successfully circumvented the problem of paper and salt noise. On the other hand, in order to deal with the picture that has been flipped, we must first establish whether the image has been flipped or not; this is a completely different approach from the one that we discussed in our main study. EasyOCR was unable to identify about one third of the second dataset as a consequence of this limitation. As a direct result of this, the confidence score is lower than 50%. Nevertheless, for the same reason, we end up with lower scores in each and every one of YOLO's variations.

After the detection procedure, a confidence score of 68.43% is achieved by using OCR technology to read and extract text from the LPs. Incorporating this kind of technologies into multimedia-cloud storage [20], [21], [45], [60], especially within a resource-constrained environment not only advances the domains of transportation safety and law enforcement in the developing countries but also holds significant importance in bolstering the safety of vehicle passengers. This concern is particularly noticeable in the cities of Bangladesh, as reported by a recent study [53]. On the other hand, Gnanaprakash et al.[40] conducted a study on automatic number plate recognition using deep learning. The obtained accuracy of their optical character detection was 96.7%. In another research Atikuzzaman et al. [30] detected and categorized vehicle number plate using CNN. The recognition accuracy of their proposed method is 91.38%. Also, many other papers [18], [31], [40], [50] did not

show any quantitative evaluation of their work by providing the comparisons of SSIM scores, PSNR scores etc. Whereas we have shown the SSIM score comparison, PSNR score comparison, NIQE values in order to provide the quantitative evaluation of our work. By comparing with others it can be seen that our study not only provides a new way to detect and recognize Bangla car plate but also has achieved a better accuracy.

Chapter 9

Conclusion

In our study, we have used the DCP dehazing method in conjunction with the state-of-the-art YOLOv8 object detection model to provide a new ALPD strategy optimized for challenging fog. Due to its low-resource status, the Bangla language is generally disregarded when considering digital text difficulty; this study aims to enhance the detection and identification of Bangla automobile plates. A crucial element in guaranteeing the visibility and accuracy of LP identification is the DCP algorithm's ability to reduce fog, which successfully improves picture clarity. From the dehazed photos, YOLOv8 offers a very efficient way to recognize Bangla LPs. Impressively, this method has achieved an LP recognition maximum accuracy of 98.5% for dataset-1 and for dataset-2 we get 80% accuracy. After the detection procedure, deciphering and extracting text from LPs is done using OCR technology, with a maximum confidence score of 68.43% for dataset-1 and 34.02% for dataset-2.

Moreover, our earlier work “Fog-Resilient Bangla Car Plate Recognition using Dark Channel Prior and YOLO” [66] at the Workshop on Vision and Language (WVLL) at the Winter Conference on Applications of Computer Vision (WACV) was approved and expanded upon in this journal submission. This study's more extensive analysis and extended findings are based on the earlier findings.

Bibliography

- [1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [2] X. Wang, “Laplacian operator-based edge detectors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 886–890, 2007.
- [3] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” 2009, Online. Accessed: 15-Sep-2023. [Online]. Available: https://projectsweb.cs.washington.edu/research/insects/CVPR2009/award/hazeremv_drkchnl.pdf.
- [4] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [5] B. Xie, F. Guo, and Z. Cai, “Improved single image dehazing using dark channel prior and multi-scale retinex,” in *2010 International Conference on Intelligent System Design and Engineering Application*, IEEE, 2010.
- [6] X. Dong *et al.*, “Fast efficient algorithm for enhancement of low lighting video,” in *2011 IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.
- [7] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (alpr): A state-of-the-art review,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, 2013.
- [8] X. Jiang, H. Yao, S. Zhang, X. Lu, and W. Zeng, “Night video enhancement using improved dark channel prior,” in *2013 IEEE International Conference on Image Processing*, 2013, pp. 553–557.
- [9] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, “Efficient image dehazing with boundary constraint and contextual regularization,” in *2013 IEEE International Conference on Computer Vision*, IEEE, 2013.
- [10] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a ‘completely blind’ image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.
- [11] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv [cs.CV]*, 2015.

- [13] S. Azam and M. M. Islam, "Automatic license plate detection in hazardous condition," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 172–186, 2016.
- [14] D. Berman, T. Treibitz, and S. Avidan, "Non-local image dehazing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 1674–1682.
- [15] B. Cai *et al.*, "Dehazenet: An end-to-end system for single image haze removal," *arXiv preprint arXiv:1601.07661*, 2016. [Online]. Available: <http://arxiv.org/abs/1601.07661>.
- [16] F. A. Fardo, V. H. Conforto, F. C. de Oliveira, and P. S. Rodrigues, "A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms," *ar5iv*, 2016.
- [17] S. Lee, S. Yun, J.-H. Nam, C. S. Won, and S.-W. Jung, "A review on dark channel prior based image dehazing algorithms," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, 2016.
- [18] M. I. Anwar and A. Khosla, "Vision enhancement through single image fog removal," *Engineering Science and Technology, an International Journal*, vol. 20, no. 3, pp. 1075–1083, 2017.
- [19] C. Godard, O. Mac, A. Gabriel, and J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Godard_Unsupervised_Monocular_Depth_CVPR_2017_paper.pdf.
- [20] J. Noor, H. I. Akbar, R. A. Sujon, and A. A. Al Islam, "Secure processing-aware media storage (spms)," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, IEEE, 2017, pp. 1–8.
- [21] J. Noor and A. A. Al Islam, "Ibuck: Reliable and secured image processing middleware for openstack swift," in *2017 International Conference on Networking, Systems and Security (NSysS)*, IEEE, 2017, pp. 144–149.
- [22] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804 [cs.CV]*, 2017.
- [23] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.
- [24] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6612–6619.
- [25] S. Abdullah, M. M. Hasan, and S. M. S. Islam, "Yolo-based three-stage network for bangla license plate recognition in dhaka metropolitan city," in *2018 International Conference on Bangla Speech and Language Processing (ICB-SLP)*, 2018, pp. 1–6.
- [26] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Benchmarking single image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2018.

- [27] C. Luo and et al., “Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding,” *arXiv preprint arXiv:1808.03867 [cs.CV]*, 2018.
- [28] N. M. Shatnawi, “Bees algorithm and support vector machine for malaysian license plate recognition,” *International Journal of Business Information Systems*, vol. 28, no. 3, p. 284, 2018.
- [29] J. Wang, K. Lu, J. Xue, N. He, and L. Shao, “Single image dehazing based on the physical model and msrnr algorithm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2190–2199, 2018. DOI: 10.1109/TCSVT.2017.2755942.
- [30] M. Atikuzzaman, M. Asaduzzaman, and M. Z. Islam, “Vehicle number plate detection and categorization using cnns,” in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2019.
- [31] M. Atikuzzaman, M. Asaduzzaman, and M. Z. Islam, “Vehicle number plate detection and categorization using cnns,” in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2019.
- [32] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3828–3838. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/papers/Godard_Digging_Into_Self-Supervised_Monocular_Depth_Estimation_ICCV_2019_paper.pdf.
- [33] M. M. S. Rahman, M. Mostakim, M. S. Nasrin, and M. Z. Alom, “Bangla license plate recognition using convolutional neural networks (cnn),” in *2019 22nd International Conference on Computer and Information Technology (IC-CIT)*, 2019, pp. 1–6.
- [34] N. Saif *et al.*, “Automatic license plate recognition system for bangla license plates using convolutional neural network,” in *2019 IEEE Region 10 Conference (TENCON)*, IEEE, 2019.
- [35] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020.
- [36] R. Islam, M. R. Islam, and K. H. Talukder, “An efficient method for extraction and recognition of bangla characters from vehicle license plates,” *Multimed. Tools Appl.*, vol. 79, no. 27-28, pp. 20 107–20 132, 2020.
- [37] N.-A. Alam, M. Ahsan, M. A. Based, and J. Haider, “Intelligent system for vehicle number plate detection and recognition using convolutional neural networks,” *Technologies (Basel)*, vol. 9, no. 1, p. 9, 2021.
- [38] K. Durden, *Bangladeshi License Plate Object Detection Dataset (v1, lpv1)*, Roboflow Universe, Generated on Nov 16, 2021, 2021. [Online]. Available: <https://universe.roboflow.com/kaiser-durden/bangladeshi-license-plate-fonfq/dataset/1>.
- [39] Z. Fang, Q. Wu, D. Huang, and D. Guan, “An improved dcp-based image defogging algorithm combined with adaptive fusion strategy,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–13, 2021.

- [40] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, “Automatic number plate recognition using deep learning,” *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012 027, 2021.
- [41] A. P. Gulati. “Text detection from images using easyocr: Hands-on guide.” Accessed on 15-Sep-2023, Analytics Vidhya. (Jun. 2021).
- [42] X. Jin, R. Tang, L. Liu, and J. Wu, “Vehicle license plate recognition for fog-haze environments,” *IET Image Processing*, vol. 15, no. 6, pp. 1273–1284, 2021.
- [43] A. Kumari, S. K. Sahoo, and M. C. Chinnaiyah, “Fast and efficient visibility restoration technique for single image dehazing and defogging,” *IEEE Access*, vol. 9, pp. 48 131–48 146, 2021.
- [44] M. A. A. Nasim, A. I. Chowdhury, J. N. Muna, and F. M. Shah, “An automated approach for the recognition of bengali license plates,” in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021, pp. 1–4.
- [45] J. Noor, S. I. Salim, and A. A. Al Islam, “Strategizing secured image storing and efficient image retrieval through a new cloud framework,” *Journal of Network and Computer Applications*, vol. 192, p. 103 167, 2021.
- [46] A. Sams and H. H. Shomee, *Bangla lpd - a*, version v1, Zenodo, Apr. 2021. DOI: 10.5281/zenodo.4718238. [Online]. Available: <https://doi.org/10.5281/zenodo.4718238>.
- [47] X. Wang, J. Liu, and X. Zhu, “Early real-time detection algorithm of tomato diseases and pests in the natural environment,” *Plant Methods*, vol. 17, no. 1, p. 43, 2021.
- [48] A. Ashrafee, A. M. Khan, M. S. Irbaz, and M. A. A. Nasim, “Real-time bangla license plate recognition system for low resource video-based applications,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2022.
- [49] IJRASET. “Vehicle number plate recognition system using tesseract-ocr.” (2022).
- [50] V. N. More and V. Vyas, “Removal of fog from hazy images and their restoration,” *Journal of King Saud University - Engineering Sciences*, 2022.
- [51] X. Nie, Z. Xu, W. Zhang, X. Dong, N. Liu, and Y. Chen, “Foggy lane dataset synthesized from monocular images for lane detection algorithms,” *Sensors*, vol. 22, no. 14, p. 5210, 2022.
- [52] A. A. J. Pazhani and S. Periyannayagi, “A novel haze removal computing architecture for remote sensing images using multi-scale retinex technique,” *Earth Sci. Inform.*, vol. 15, no. 2, pp. 1147–1154, 2022. DOI: 10.1007/s12145-021-00610-z.
- [53] A. B. Alok *et al.*, “khep’: Exploring factors that influence the preference of contractual rides to ride-sharing apps in bangladesh,” in *Proceedings of the 6th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies*, 2023.

- [54] Bangladesh Road Transport Authority. “Number of Registered Vehicles in Whole BD.” Accessed on May 20, 2023. (2023), [Online]. Available: <http://www.brta.gov.bd/site/page/74b2a5c3-60cb-4d3c-a699-e2988fed84b2/Number-of-registered-Vehicles-in-Whole-BD>.
- [55] Datacamp.com. “Yolo object detection explained.” (2023), [Online]. Available: <https://www.datacamp.com/blog/yolo-object-detection-explained>.
- [56] C. V. Engineer. “Train yolov8 object detection on a custom dataset — step by step guide — computer vision tutorial.” Accessed on 18-Sep-2023. (Jan. 2023).
- [57] A. Gullbadhar. “Introduction to easyocr: A simple and accurate python library for optical character recognition.” Accessed on 15-Sep-2023, Level Up Coding. (Feb. 2023).
- [58] C. Li *et al.*, “Single-image dehazing based on improved bright channel prior and dark channel prior,” *Electronics (Basel)*, vol. 12, no. 2, p. 299, 2023.
- [59] A. Mehra. “Understanding yolov8 architecture, applications & features.” Accessed on 18-Sep-2023, Labellerr. (Apr. 2023).
- [60] J. Noor, M. N. H. Shanto, J. J. Mondal, M. G. Hossain, S. Chellappan, and A. B. M. A. Al Islam, “Orchestrating image retrieval and storage over a cloud system,” *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1794–1806, 2023. DOI: 10.1109/TCC.2022.3162790.
- [61] Y. Ouyang, W. Chai, J. Ye, D. Tao, Y. Zhan, and G. Wang, “Chasing consistency in text-to-3d generation from a single image,” *arXiv*, 2023. [Online]. Available: <http://arxiv.org/abs/2309.03599>.
- [62] Y. Shambharkar, S. Salagrama, K. Sharma, O. Mishra, and D. Parashar, “An automatic framework for number plate detection using ocr and deep learning approach,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.
- [63] H. Shi and D. Zhao, “License plate recognition system based on improved yolov5 and gru,” *IEEE Access*, vol. 11, pp. 10 429–10 439, 2023.
- [64] J. Solawetz. “What is yolov8? the ultimate guide.” Accessed on 18-Sep-2023, Roboflow Blog. (Jan. 2023).
- [65] K. Moreland, *Diverging color maps for scientific visualization (expanded)*, <https://www.kennethmoreland.com/color-maps/ColorMapsExpanded.pdf>, 2024.
- [66] H. I. Nasim, F. J. Printia, M. Hasan, *et al.*, “Fog-resilient bangla car plate recognition using dark channel prior and yolo,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, Jan. 2024, pp. 1110–1119.