Object Detection under Diverse Weather Conditions for Autonomous Systems using Visual Prompting

by

Sreya Majumder
21201742
Syed Faysel Ahammad Rajo
21101078
Arundhati Sarkar Sudipta
23241077

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 and Engineering

Department of Computer Science and Engineering
Brac University
October 2025

Declaration

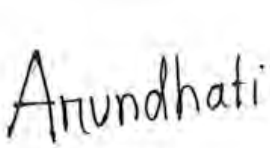
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sreya Majumder
21201742

Syed Faysel Ahammad Rajo
21101078



Arundhati Sarkar Sudipta
23241077

Approval

The thesis/project titled "Object Detection under Diverse Weather Conditions for Autonomous Systems using Visual Prompting" submitted by

1. Sreya Majumder(21201742)
2. Syed Faysel Ahammad Rajo(21101078)
3. Arundhati Sarkar Sudipta(23241077)

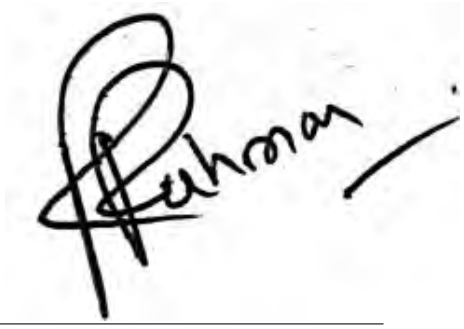
Of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 10, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Mr. Rafeed Rahman
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Object detection plays a vital role in enabling autonomous systems to perceive their surroundings and make effective decisions while navigating in complex environments. In recent years, remarkable advancements have been achieved in this field by leveraging pretrained models. However, the object detection process across diverse weather remains a challenge for these pretrained models as different weather conditions introduce visual distortions in the images. To address this problem, we propose a novel approach by combining visual prompting with fine tuning of real time object detection models to improve the detection accuracy. Visual prompting enables lightweight input level adaptation by introducing learnable prompts which adjust the input representation during training time by adding only a small number of parameters. In this research, visual prompting was integrated with the YOLOv8 model and the modified YOLOv8 model with ResNet50 backbone. The models were evaluated on five distinct weather datasets including daytime-clear, daytime-foggy, night-clear, dusk-rainy and night-rainy conditions. where the experimental results demonstrated significant improvement in different weather conditions for both models. Thus this research provides an efficient strategy to improve the perception capabilities of autonomous systems in challenging environments.

Keywords: Object Detection; Visual prompting; Autonomous Systems

Acknowledgement

Firstly, all praise to the Almighty for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Md. Golam Robiul Alam for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, to our cosupervisor Mr. Rafeed Rahman for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

AI Artificial Intelligence

AV Autonomous Vehicle

BDD Berkeley Deep Drive

CNN Convolutional Neural Network

CSP Cross Stage Partial

DBB Diverse Branch Block

DWD Diverse Weather Dataset

FGVP Fine Grained Visual Prompting

FLOPs Floating Point Operations

FPN Feature Pyramid Network

FP False Positives, incorrectly predicted object instances.

FN False Negatives, missed object instances.

FR Faster R-CNN

GELAN Generalized Efficient Layer Aggregation Network

GOT Generic Object Tracking

GPT Generative Pre-trained Transformer

GT Ground Truth

IBN-Net Instance-Batch Normalization Network

ICL In-Context Learning

IoU Intersection over Union

IterNorm Iterative Normalization

JS Jensen-Shannon (divergence)

mAP mean Average Precision

mAP50 Mean Average Precision at IoU threshold 0.5

mAP50-95 Mean Average Precision at IoU thresholds from 0.5 to 0.95

NMS Non-Maximum Suppression

PAN Path Aggregation Network

PGI Programmable Gradient Information

PGN Prompt Generation Network

PHOW Pyramid Histogram of Words

R-CNN Region-based Convolutional Neural Network

ResNet50 A 50-layer Residual Network architecture

RGB Red Green Blue (color channels)

RM Relation Modeling

RPN Regional Proposal Network

RT-DETR Real-Time Detection Transformer

SVM Support Vector Machine

SW Switchable Whitening

TP True Positives, correctly predicted object instances.

TPR Test-Time Prompt Refinement

VIMA Vision-Language Model for robot manipulation

VP Visual Prompt

VRP Vehicle Routing Problem

XML Extensible Markup Language

YOLO You Only Look Once

YOLOv8 You Only Look Once version 8

Chapter 1

Introduction

1.1 Background

Autonomous systems are the systems that are capable of doing tasks and changing behavior according to their situation without human intervention.[29] In the modern world, due to the rapid advancements of these systems such as self-driving cars, robots, drones are increasingly transitioning the industries, agriculture, medical sectors and enhancing daily life. These systems need to understand their environments, detect objects and make decisions in real time. In recent years, these systems have shown significant progress due to the rapid advancements of artificial intelligence, deep learning techniques and computer vision. However, in spite of the advancements, they still struggle in dynamic environments where they need to operate in different weather, lighting and unstructured environments. A key challenge of these systems is detecting objects accurately in diverse weather while maintaining their real time performance.

Detecting objects based on the scenario has been a constant challenge for autonomous systems and several methods have been used to address this challenge. Traditionally, Support Vector Machines(SVMs), PHOW (Pyramid Histogram of Words) methods have been used for object detection which did not perform well in detecting objects under diverse weather scenarios.[3] To make the performance better, deep learning approaches have played a pivotal role. Convolutional Neural Network(CNN) based architectures have been extensively used for detecting objects efficiently in real time. According to [43], CNN based double stage and single stage detector architectures such as Faster R-CNN, YOLO have shown significant success where the double stage detectors increase the accuracy and single stage detectors maintain the speed. These models are trained on large dataset which consists of different classes. In order to use these models for object detection, these models are fine tuned on a specific dataset. Even though the models can provide a balanced performance in clear weather, the accuracy of these autonomous vehicles significantly drop under adverse weather conditions.

In recent years, visual prompting has emerged as a technique to adapt pretrained vision models to a specific downstream task by changing the input pixel [12]. This method has gained attention due to its success in adapting large pretrained models in a parameter efficient way. Various research conducted on this topic have demon-

strated its success in classification tasks [28][17]. However, very few studies have explored this technique for object detection. So, it is important to analyze the performance of visual prompting in object detection, specifically in a diverse weather context where the performance of the state of the art(SOTA) models degrade as the weather and lighting change affect the images severely.

1.2 Motivation

Over the past few years, the demand for autonomous systems in various sectors have increased. According to [43], 73% of total deaths are caused by traffic accidents which has increased the importance of developing autonomous vehicles like self-driving cars. Along with that, the need for autonomous drones and robots have also increased for sectors ranging from surveillance to agriculture [5]. These autonomous systems need to deal with varying lighting and weather conditions as they need to perform in different countries having different environmental conditions. However, these systems still struggle a lot in detecting objects correctly in complex scenarios. As a result, improving the detection performance of the autonomous systems becomes very important. Moreover, visual prompting has also received substantial attention due to its adaptation to vision models [12]. So, the utilization of visual prompting in object detection specifically in diverse weather contexts remains an underexplored area with significant potential for improving object detection performance.

1.3 Problem Statement

Despite significant advancements of deep learning models in object detection, existing models face significant challenges in detecting objects accurately. The difficulty increases when the model needs to detect objects in dynamic environments with different weather and lighting conditions [13][46][48]. To address these issues, several methods including image restoration, denoising, and augmentation have been explored along with the standard fine-tuning approach. However, these techniques often significantly increase the computational complexity and inference time [35] [43]. Therefore, there is a need for lightweight, efficient, and adaptable methods that can improve the object detection performance and robustness of the models. Various recent studies have shown that visual prompting can improve the performance of pretrained models while introducing a small number of parameters [12]. However, the mechanism has mostly been used in adapting large pretrained classification models [28][17]. Very few studies have explored the technique to improve object detection performance[36]. As a result, the application of visual prompting in object detection under diverse weather conditions remains largely underexplored. This research seeks to investigate the effect of visual prompting technique when combined with the standard fine tuning process of lightweight object detection models, in order to enhance the detection performance, particularly in diverse weather.

1.4 Research Objectives

This research aims to integrate visual prompting mechanism into object detection process in order to improve the perception and the decision-making capabilities of autonomous systems under diverse weather conditions.

The research objectives are:

- **To propose** a Visual Prompting (VP) integrated fine-tuning method for pre-trained models in the context of object detection under diverse weather conditions.
- **To investigate** the impact of incorporating VP into both the standard YOLOv8 model and the modified YOLOv8 model with ResNet50 backbone.
- **To compare** the performance of VP based fine tuning with the standard fine tuning approach, assessing difference in various evaluation metrics like detection accuracy.
- **To examine** whether the backbone choices affect the effectiveness of visual prompting by comparing two YOLOv8 variants using default and ResNet50 backbone.
- **To identify** the limitations of the proposed method and provide recommendations for future work.

1.5 Methodology in Brief

The main goal of this research is to implement a system in order to increase the efficiency of the object detection process under different weather conditions. So, the first step was collecting data containing both different weather and lighting conditions. The data included image and annotation files. After that, the raw image data and the annotation files were prepared for training the models using various preprocessing techniques. Then the data was splitted for each weather. Further, the baseline model YOLOv8 and the modified model with ResNet50 backbone was implemented. Then the models were fine tuned on each weather and the performance of these models were analyzed. After that, the visual prompt was designed and integrated with both of the models. Then the VP integrated models were trained on each weather and the performance of these enhanced models were analyzed. After that, the performance of these models were compared with the baseline methods.

1.6 Scopes and Challenges

1.6.1 Scopes

Our proposed methodology can be used in various real world applications which need to deal with different outdoor environments. While the research mainly emphasizes driving scenarios under diverse weather, the method can be applied to other autonomous systems as well. For instance, autonomous robots need to work in various sectors including search and rescue, agriculture, surveillance. So, they need to focus on objects in scenarios when occlusions and varying lighting conditions are present. Moreover, autonomous drones need to detect objects correctly in various outdoor environments. As a result, they need to deal with adverse weather conditions and detect objects accurately. The perception capability of these autonomous systems get affected due to the adverse weather conditions. So, our method can be implemented for these systems in order to enhance their detection performance while maintaining real time performance.

1.6.2 Challenges

During the research, we faced several challenges. To begin with, collecting diverse weather data with different objects was a big challenge as a few datasets were available containing diverse weather scenarios. Moreover, the integration of visual prompting with the YOLOv8 model and the modified model was challenging as the integration required low level customization. Along with that, the modification of backbone architecture was quite difficult as the other structure of other parts of the model were kept preserved. Further, this thesis required a lot of training processes to analyze the performance of different methods across different weather conditions. As a result, the process became time-consuming even after using a high-end GPU.

Chapter 2

Literature Review

2.1 Preliminaries

The emergence of visual prompting strategies has introduced a new research direction. Prompting, which was originally popularized in natural language processing, enables models to adapt to downstream tasks with minimal retraining. In computer vision, visual prompting and parameter-efficient tuning methods such as CoOp, CoCoOp, and Visual Prompt Tuning (VPT) have shown promise in classification and recognition tasks [15]. However, their application to object detection, particularly in challenging weather conditions, remains underexplored. This literature review examines prior research across three major dimensions: (i) the evolution of object detection techniques, (ii) approaches addressing weather-induced challenges in detection, and (iii) recent developments in visual prompting and their adaptation to computer vision. By critically analyzing these domains, the review identifies existing gaps and highlights the need for exploring visual prompting in object detection models such as YOLO[7] when applied to weather-diverse datasets like DWD.

2.1.1 Object Detection in Computer Vision

Object detection is a core computer vision task that consists of object localization and classification in an image. Initial approaches were based on handcrafted characteristics, including Haar cascades, Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) with conventional classifiers, including Support Vector Machines (SVMs) [41]. Though efficient within restricted settings, these methods were not robust and scaled in challenging and real-life settings. Convolutional neural networks (CNNs) were introduced, which triggered a paradigm shift in object detection. The R-CNN family was the first to utilize region-based methods with R-CNN introducing selective search as candidate regions and then CNN-based classification. Its successors, Fast R-CNN and Faster R-CNN were more efficient, combining feature extraction and region proposal generation, and greatly lowering the computational load[14].
Simultaneously, one-stage detectors were introduced to remove the usage of region proposal networks. You Only Look Once (YOLO) [7] framework and Single Shot MultiBox Detector (SSD) were able to perform real-time detection formulated detection as the regression problem. Specifically, YOLO has been developed over several iterations (v1-v8), with each one adding accuracy, speed, and feature representa-

tion, but retaining real-time performance. Another interesting development was RetinaNet which added the concept of focal loss to deal with the problem of class imbalance. More recently, models that rely on transformers like Detection Transformer (DETR) have expanded the set of tasks that can be accomplished with object detection to include global reasoning via self-attention. These algorithms have good performance, yet tend to be computationally expensive, and therefore lightweight and efficient detectors like YOLO are of interest to resource-constrained or real-time applications.

2.1.2 Existing Methods for Handling Weather Challenges

In order to minimize the negative effect of weather in object detection researchers have proposed some strategies as follows:

- **Data Augmentation:** Synthetic weather effects such as simulated fog, rain or snow are applied to training datasets to increase model robustness [49]. Despite their usefullness, these methods are not typically able to reflect the richness of nature conditions.
- **Domain Adaptation and Generalization:** The domain adaptation methods aim at training in clear-weather conditions and transfer the knowledge to adverse weather conditions.
- **Specialized Architectures:** Other suggestions include weather-aware modules, attention, or multimodal fusion (e.g. LiDAR or radar) to aid in detection in poor conditions. However, these techniques are more likely to increase the computational costs and reduce the feasibility in real time.

Along with these advances, there has been a tendency to either utilize synthetic weather augmentation or highly engineered models. Minimal research has been done to explore prompting-based methods to obtain robustness to weather variations.

2.1.3 Visual Prompting

Visual prompting is a developing idea that is based on the effectiveness of prompting in natural language processing (NLP) [24][42]. Prompts in NLP are applied to instruct large pretrained language models to do specific downstream tasks without adjusting their internal parameters [15]. Applying this concept to the vision domain, visual prompting allows such an adaptation to apply the same kind of input-level changes instead of retraining or finetuning the entire model [12]. The main concept is to overlay a little number of learnable parameters- called visual prompts- directly onto the input image, e.g. patches, masks, or even embeddings. These parameters affect the model feature extraction process either during training or inference. Visual prompting enables models to optimize effectively on new tasks or domains at low computational cost as compared to conventional fine-tuning methods that involve tuning millions of parameters [42]. The process preserves the generalization capability of the pretrained model and enhances its performance on specialized data. An example is when the environmental factors like rain, fog or low light deteriorate

the quality of the image, visual cues can be used to make the model re-calibrate its perception by slightly shifting focus to pertinent areas or to eliminate color or texture biasness. Recent research has established that visual prompting can be used successfully to other computer vision problems such as image classification, semantic segmentation, and object detection. It is lightweight and flexible, which makes it especially applicable to situations where retraining of large models is not feasible. Visual prompting provides a potential way forward in improving the robustness and adaptability of vision models by allowing task specific adaptation by manipulation of inputs, particularly in difficult real world situations like weather variations.

2.2 Related Works

Gupta et al. (2024) introduced a robust method for object detection under challenging weather conditions [35]. It addressed a critical limitation in vision-based systems such as autonomous driving and outdoor surveillance. The authors compared three distinct strategies such as (1) training directly on real-world all-weather images, (2) using synthetic weather augmentation and (3) applying image-denoising before detection. They experimented using YOLOv5 on the BDD100K dataset and evaluated performance on unseen adverse-weather images from DAWN. Their results show that training on real all-weather data consistently outperforms augmentation-based strategies. However, integrating image denoising before object detection in adverse weather scenarios results in poorer detection performance compared to models trained directly on noisy images. Gupta et al.'s work is notable because it doesn't presume any single solution but rather introduced a comparative empirical evaluation of multiple weather-robustness strategies. Their key finding that denoising usually degrades detection accuracy highlights the difficulty of balancing visual quality and performance of downstream tasks. This study forms a useful baseline and contrast point for our work that instead explores input-level adaptation using visual prompting.

In this paper [11], Bahng et al. explore visual prompting as a mechanism to adapt large-scale vision models to new tasks without fully fine-tuning their internal weights. They frame prompting as learning a single image perturbation (i.e., a learnable input-level modification). When appended to inputs of a frozen pretrained model, it helps the model perform on a downstream task. In their experiments, visual prompting proved especially effective when applied to CLIP and showed robustness under domain shifts. The authors also analyze the effects of prompt design, dataset features and output transformation on adaption performance. It shows that relatively simple, trainable input modifications can result in significant adaptability in pretrained vision models. In relation to our work, their findings provide theoretical support for using input-level adaptations like visual prompt integration rather than heavy fine-tuning.

In the paper [13], "Object Detection in Autonomous Vehicles" (IEEE, 2023), the authors discuss how object detection under diverse weather or lighting conditions is crucial for autonomous vehicles to ensure road safety. The study mainly focuses on the YOLO (You Only Look Once) model and how it compares with several other deep learning-based detectors such as RetinaNet, Fast R-CNN, and SSD. Based on

their experiments and analysis, YOLO stands out for its faster inference speed and competitive accuracy. That is why YOLO is suitable for real-time driving scenarios where quick decision-making is important. The paper also emphasizes that YOLO's single-stage detection pipeline helps reduce computational overhead without compromising too much on precision. However, the authors did not explore recent adaptation strategies such as visual prompting. Their findings clearly highlight the strength of YOLO-based approaches as a reliable foundation for object detection tasks.

Tahir et al. (2024) provide an extensive overview of existing tools in object detection, both traditional and deep learning-based approaches, in relation to the case of autonomous vehicles operating in poor weather conditions [43]. They investigate the effects of precipitation, fog, snow, low lighting, and other adverse environmental factors on detection in a systematic manner, and point out how standard techniques such as HOG, SIFT, Viola-Jones and SVM-based classifiers have historically failed because of their use of hand-designed features and their inability to adapt to changing visual distortions. Conversely, the authors claim that deep neural networks, particularly convolutional networks, are more resistant to environmental changes because the hierarchical feature representations are learned directly on the data. They categorize object detectors as one-stage and two-stage methods, elaborate on popular models such as Faster R-CNN and YOLO, and trade-offs between speed and accuracy where hybrid approaches can alleviate each weakness. In addition to algorithmic comparison, the review also provides an overview of the benchmark datasets often utilized in adverse-weather object detection studies (e.g. KITTI, Waymo, ACDC, NuScenes) and describes popular evaluation metrics. The survey offers a useful base and guide to creatively building more capable detection systems filling gaps, including lack of coverage of the less-understood weather types and small or partially hidden object-detecting.

Li et al. (2023) introduced a framework for visual in-context prompting which focuses at improving image segmentation tasks for both generic and referring segmentation [22]. Segmentation often relies on predefined labels or text-based prompts in traditional models, but the authors propose a system by developing a prompt encoder that supports in-context visual prompts such as strokes, masks, or boxes which allows the model to segment objects more efficiently across various tasks, including open-set detection and segmentation. Their model DINOv [22], which is built on an encoder-decoder architecture, handles multiple types of visual prompts, adapting them to specific tasks and objects in a given image. The ability of their model DINOv to support both generic and referring segmentation within a single framework, enabling it to generalize well even on open-set tasks, where new, unseen objects are presented distinguishes it from other previous approaches such as SegGPT[30]. Their model performed significantly on benchmarks like COCO and ADE20K shows great results in generalizing from a limited set of training data, considering existing models in both closed-set and open-set segmentation. The visual prompting system is the main purpose of DINOv.This approach differs significantly from visual perception models based on the use of texts, such as CLIP [10], which employ texts to guide the segmentation. Instead, Li et al. claim that visual prompts instead of text prompts are much better in dealing with complex image segmenta-

tion problems where the text prompt can be ambiguous. This is consistent with the ongoing attempts to build computer vision foundation models such as the Segment Anything Model (SAM)[20].

The Segment Anything Model (SAM), as described in [20] represents a significant enhancement progression from basic image segmentation towards a concept of a general segmentation task. SAM proposes a promptable segmentation framework to endow the model with the ability of generalizing to diverse segmentation tasks through prompt engineering. Unlike other segmentation models for specific tasks, SAM is specially designed so that it can be fed with arbitrary prompts (e.g. points, boxes, and masks), and gracefully deal with ambiguous inputs, a capability required for real time interactive segmentation. This work contributes to one of the key contributions to the problem, the SA1B dataset for multiclass segmentation, that contains over 1.1 billion segmentation masks obtained via an iterative process of manual, semi automatic and fully automatic annotation. As the scale and diversity of our dataset, we can generalize in zero shot, and outperform previous work on a family of different benchmarks. Same as CLIP [10], SAM is a strong candidate for foundation models in computer vision due to its ability to generate high quality segmentation masks conditioned purely on only a few user input. Additionally, the authors demonstrate that SAM is flexible across tasks not in its training, featuring zero shot transfer. SAM is thus positioned as a tool for image understanding, bridging the gap between supervised learning and the recent mode of self-supervised models. The release of SAM along with the SA-1B dataset promises to inspire future innovations of computer vision research by creating an effective and scalable solution to the segmentation tasks. This work is a key step forward in the capabilities of prompt based segmentation, making it possible for models to handle open world scenarios and interact with real time users, critical advancements for interactive vision systems.

The PANDA [25] (prompt-based context and domain-aware pretraining) framework by Liu et al.(2023) is implemented to improve vision and language navigation (VLN) by enhancing pre-trained models to understand context, such as indoor environments. PANDA combines both context-aware and domain-aware prompting techniques, which enable agents to make sequential decisions in photo-realistic indoor environments. The authors present an approach that helps VLN agents or systems in comprehending both scene semantics and action sequences. The research focuses on two major issues: 1) Pretrained models lack sensitivity to indoor environments, resulting in poor performance in VLN tasks; 2) Sequential decision-making in VLN[25] tasks is hindered due to a lack of understanding of the relationships between navigational actions. The model utilizes deep visual prompts and hard context prompts. From indoor datasets visual prompts are learned. It makes pretrained models sensitive to indoor environments. The R2R and REVERIE datasets are used for training and evaluation that focuses on aligning action sequences with navigation instructions. PANDA's performance depends on its ability to precisely align action sequences with navigational paths. It currently struggles to generalize across unique unseen environments. Also, its adaptability to environments with dif-

ferent topologies is limited due to its reliance on prompt tuning.

The authors Rao et al.(2022) came up with a framework called DenseCLIP [16], that applies the CLIP[10] model to dense prediction tasks such as semantic segmentation and object detection. It focuses on converting the image-text matching task of CLIP into pixel-text matching in order to assist dense prediction tasks with context-aware prompting for improved accuracy. Their model addresses the challenge of using CLIP's image-text knowledge for dense prediction tasks. It connects the gap between instance-level tasks (such as classification) and pixel-level tasks (such as segmentation) that CLIP's original formulation struggled with. DenseCLIP modifies CLIP's architecture by incorporating pixel-text score maps and employs context-aware prompting to optimize the pre-trained model for dense prediction. This model is evaluated using the ADE20K dataset for semantic segmentation and the COCO dataset for object detection and instance segmentation. DenseCLIP works well with segmentation tasks, but its advances in object detection are limited. Additionally, while it improves pixel-level understanding, it still significantly depends on pre-trained vision-language models (VLMs) that restricts its applicability to increasingly complicated, previously unknown data sets.

The Fine Grained Visual Prompting (FGVP)[32] framework from Yang et al. (2023) was introduced to increase accuracy in instance level tasks of Vision Language Models (VLMs). This paper applies blur based techniques and semantic masks to zero shot performance on tasks such as part detection and expression understanding using several visual prompting methods. The research attempts to solve the problem of the lack of fine grained precision in current methods of visual prompting, which are based on using rough visual markers such as boxes or circles. However, these methods often include irrelevant pixels that prevent achieving sub optimal localization and recognition. For precise prompting of the visual component, FGVP relies on Blur Reverse Masks, which have been generated using SAM, the Segment Anything Model. For accurate visual prompting, FGVP utilizes Blur Reverse Masks from the Segment Anything Model (SAM)[20]. To assess the model's ability to perform zero shot image comprehension tasks, it is tested using a variety of datasets including RefCOCO, RefCOCO+, RefCOCOg and PACO. Inference time of the method is longer because the method requires a long time to generate accurate semantic masks. Additionally, little investigation is made on text-image alignment by FGVP, which could improve the performance not only only on some particular tasks, but also on the whole system.

The authors describe a multimodal prompt (text and image) based framework for robot manipulation called VIMA [18]. It is presented as a single, adaptable learning interface for robots that handles various task types (visual goals, language instructions, and demonstrations) within a single unified framework. VIMA takes on the problem of specialization in robotic systems, which often requires separate models for each task. A unified prompting mechanism is used to learn and perform multiple tasks while improving generalization across previously unseen tasks in a single model. The model uses a transformer-based model with cross-attention that ties task specific prompts to robot actions. This research presents VIMABENCH based

on the Ravens simulator with 17 tasks and about 600K expert trajectories for imitation learning. Despite performing well on new and more challenging tasks (Level 4), it has difficulties generalizing to new tasks and are prone to errors relying on object detectors like Mask R-CNN. Yet data augmentation alleviates errors in detection.

In this research article [31], Wang et al. (2023) describe a new approach to object detection that is enriched with the models of vision and language, GPT-4. For that purpose visual and text prompts are used together. Object segmentation is performed by Segment Anything Model (SAM), and knowledge distillation, a process that helps create smaller models from bigger ones, is utilized in the course of the project to enhance effectiveness without notable loss of accuracy. The originality of this work consists, among other things, in the introduction of dynamic stitching, which reduces the expenses of computation by allowing processing of only those image views that are relevant to the task at hand. Such improvement also helps in the point cloud generation as after the selected views are picked, they are integrated to form a 3D image after cleaning up the noise. Because of this, the technique is capable of achieving better quality 3D models with less processing power required. Their distilled model was found by the authors to be equally effective to bigger models while doing standard evaluation and yet encouragingly at a lower cost of computation. This makes it a great solution for tasks that need both accuracy and efficiency in object detection. SAM's ability to segment objects precisely will help in your task of creating bounding boxes around objects after detecting the environment also knowledge distillation could be used to create lightweight models.

Authors of this paper[39], Ahmet, A., Allmendinger,A., Stein,A. (2024) compares three leading detectors-YOLOv9, YOLOv10 and RT-DETR-on a bespoke set of farm images featuring sugar beet, monocot and dicot weeds, testing each model size from nano to large and resizing inputs between 320 and 960 px. Authors train the networks in the Ultralytics pipeline with heavy augmentations for 250 epochs, then record mean Average Precision (mAP) alongside run times on both GPU and CPU. Findings reveal that the YOLO family-especially the newest v9 and v10-strike the best balance, scoring high mAP while processing frames much quicker than RT-DETR . Although RT-DETR still excels on fine or confused weed classes, its multi-head attention and deep decoder layers push up latency and memory demand. YOLO wins here mainly because its compact CNN, swift PGI and GELAN fusions, plus NMS-free setup, yield far lower FLOPs, fewer parameters and much faster inferences, making it better suited to budget-limited, real-time farm robots(2024) [34].

In this paper a slightly modified YOLO v8 model called YOLO-KD is used for real time road defect detection. Create a new more efficient object detection framework tailored for real time detection. The core architecture was modified in three major enhancements, first of all because road defects exhibit various complexity, it is hard for feature extraction to solve this problem they used the Diverse Branch Block (DBB) into the C2f module of YOLOv8, replacing the conventional Bottleneck. Also optimize the C2f backbone and integrate Ghost Conv and multi-scale convolution

which resolve issues with slow bounding box regression and inaccurate localization in the traditional YOLOV8 model(Sanker.,2023) [27]. Secondly, they improve localization by using IoU instead of CIoU. with CIoU it was difficult to handle different scales , complex shape and dimension of the bounding box. This loss function combines Inner-IoU (which emphasizes spatial overlap) and Shape-IoU (which considers object geometry) to improve the bounding box regression's adaptability to complex shapes. Finally, Knowledge Distillation Using KL Divergence which is the most innovative contribution that improves the detection capacity of this model. KL divergence based KD, transfers knowledge from teacher model to a lightweight student model. In the KD module, the student is trained not only on hard labels but also on the teacher's probabilistic outputs while being lightweight. In this paper, the authors use a modified version of KL divergence specifically, Jensen-Shannon (JS) divergence, which is a symmetrized and smoothed version of KL divergence. Using Kl based Knowledge Distillation significantly improves the performance compared to baseline YOLO v8 nano. The accuracy increased by 2.8 percent mAP@50. So with this method they achieved better accuracy specifically in diverse weather.

In the paper Rajendran, R., Nguyen, T., & Chen, W. (2023) [26]investigates a problem of few-shot semantic segmentation that deals with recognition of new and previously unseen classes generally by annotating only a few images of the new class. It draws on the negative to supplement the scarceness of training data in the form of few training images, both gradient-based and metric-learning-based approaches. In addition to that, transductive methods are also used by them, where the new objects are labeled with very few or no labeled images and extract features at multiple scales to improve segmentation . However, the method struggles with very complex new cases, and the transductive fine-tuning increases the time it takes to make predictions. The process starts by setting up base visual prompts and extracting features from the images at multiple scales. These base prompts are refined to segment the base classes, and then they are frozen. After that, novel visual prompts are introduced for the new classes, and a special attention mechanism links the novel prompts with the base ones. Both base and novel prompts are combined, leading to final predictions.For our research this method can enhance the accuracy of object detection and segmentation across diverse environments by ensuring the model captures details at various scales also The few-shot learning approach can help the model adapt to these without needing a large amount of labeled data

Li et al.(2023) shows pre-trained vision models to perform better in tasks by performing visual prompting but instead of focusing on one specific set of images, they work on a particular prompt in multiple sets, efficiently teaching the model how to handle diversity with ease [23]. The main improvement is segmenting as much as possible differences of the dataset, for instance lighting, angles, or complexity so that each segment of the data has its own specific visual prompt. Knowledge can be quickly adapted to a new task without having to retrain the entire model, facilitating the adoption of new tasks by the system. The overall system does not increase the amount of computation for the overall model but rather improves performance by applying targeted prompting to distinct groups of images. As other prompt-

ing techniques cannot converge efficiently and flexibly, the authors state that the technique, DAM-VP, can learn and transfer knowledge on several tasks across multi datasets and it is efficient in terms of convergence time. So we can apply DAM-VP's diversity-aware clustering that could be useful for differentiating between indoor and outdoor environments based on visual cues such as lighting, angles, and complexity. By clustering scenes and applying tailored prompts and could enhance environment recognition.

In this paper [38] Kim, J., Park, H., & Lee, S. (2024) proposes a novel technique that improves class incremental object detection using Vision-Language Models (VLMs). The method starts with pseudo-labeling, where a pre-trained model identifies potential objects in images and generates initial pseudo ground truths (GTs). This labeling is used with costume generated prompts specially for each pseudo GT, which are low quality labels to improve the consistency of the annotations. Then, these refined pseudo GTs are combined with real GTs of new tasks to form a new detector called M(new) and Then knowledge of previous tasks is combined with new tasks so that the detection of multiple objects of different classes can be improved. Nevertheless, the method suffers from limitations, the most prominent being performance drop risks from exercising the previous models during the simulation phase of labeling. Such an issue is especially worrying in sequences of multi incremental tasks wherein the task of reproducing class accuracy for learned classes in the beginning is a challenge. In our thesis, as you see targets consisting of indoor and outdoor viewpoints, pseudo-labels have the potential to accelerate the training phase for target categories that have not been encountered before or new environments.

In this paper [45], a modified In-Context Learning (ICL) model is proposed for the purpose of image segmentation by replacing the current example selection method with a more effective one. The authors solve the issues of previous works that do not utilize visual prompts; or those that select samples only based on the similarity of the image to context. The paper presents a sequential context retrieval technique that builds a limited but rich array of potential candidates and chooses appropriately fitting contextual references on a case-by-case basis. The method works towards improving the segmentation performance and at the same time decreasing the annotation burden by compressing the search space. The method addresses the issue of context dependency in the ICL based models, where a variety of prompts could potentially alter the outcome. Extensive experiments confirm the superiority of this method as compared to conventional approaches. But at the same time, the adaptive search mechanisms may lead to a higher solution cost, and their transfer to a new segmentation problem may require additional refinements. In general, the method has enabled the increase of segmentation accuracy with decreased labeling time.

This paper[21] discusses how Labellerr's advanced image segmentation method combines the Vision-Transformer-based SAM (Segment Anything Model) with Vehicle Routing Problem (VRP) optimization. The method starts by using SAM to generate an initial segmentation of the image. SAM's transformer-based architecture enables

it to handle diverse image types and accurately segment objects. To further optimize this, Labellerr integrates VRP optimization to identify and prioritize the most relevant segments within the dataset. VRP minimizes the time and effort required for labeling large image datasets by routing through high-priority annotations, improving annotation efficiency and segmentation accuracy.This method addresses the key challenges in segmenting large and complex datasets by automatically selecting the most crucial segments and reducing the need for manual labeling. However, limitations arise due to the method's reliance on pre-defined dataset characteristics, which may limit its adaptability to more diverse or generalized image segmentation tasks. For our thesis we can implement SAM directly to segment objects like indoor/outdoor scenes. It can create initial segmentations around objects in real-time, which aligns with your goal of creating bounding boxes.

The researchers of the paper [36] introduced a novel method called fine-tuned visual prompting for object detection in fisheye images. These images play an important role in capturing the expansive areas of objects, but they usually suffer from a significant radial distortion, especially at the edges. To improve the accuracy, the authors proposed a 24 point regression strategy to represent the object edges better and incorporated visual prompts to help the model focus on distorted areas of the image. They have conducted the research using Woodscape dataset which had more than 8,2300 annotated fisheye images. They first used a 24-point regression method to provide precise annotations of object boundaries. Then they included pixel level adjustment in the input image as visual prompts to guide the model for focusing on critical edges in the distorted images. After that, they used the YOLOX model which is well known for its efficiency in real-time object detection along with the visual prompting mechanism for efficient processing of 24 point regression. Additionally, they have used five pretrained CNN and transformer based vision models which are Swin transformer-tiny (Swin-T), Swin transformer-small (Swin-S), pretrained ImageNet and Darknet-53, Resnet-50 and VGG-19 pretrained COCO 2017 as backbone models. The model's performance was evaluated based on the mean Intersection over Union (IOU) and mean average precision (mAP) and it was shown that the integration of visual prompting enhanced the accuracy of detecting the edges of fisheye images. However, the computational complexity was increased due to integrating 24 point regression and the model's performance became slower because of the increased number of parameters.

Liu et al. (2024) [40] introduced a mark based visual prompting technique for robotic manipulation tasks. Traditional robotic systems often face difficulty in completing tasks while encountering unfamiliar objects or goals. In this paper, the researchers proposed a method named Marking Open-World Keypoint Affordances(MOKA) for enabling robots to perform complex manipulation tasks in open-world environments by leveraging the capability of Vision Language Model (VLM). They have conducted research on a table top manipulation environment that involves various common daily objects. Initially, a natural language instruction was given to the robotic system which was processed using a Vision-Language Model (VLM) such as GPT-4V for their ability of high level reasoning. The model then decomposed the instruc-

tion into a sequence of subtasks such as identifying the objects and the necessary actions. Further, mark based visual prompts were introduced to highlight the key points which represented the grasping points and waypoints that represented the motion trajectories. In order to select the keypoints, they used GroundedSAM combining GroundingDINO and Segment Anything Model(SAM). Furthermore, the visual prompts were used to generate a point-based affordance representation that specified the robot's movements and they simplified the complex motion tasks in visual question answering tasks by using the prompts. This method showed improved performance after comparing with two baseline methods Code-as-Policies and VoxPoser. Though this method could do complex manipulation tasks, this method had latency issues due to the long query time of VLM. In short, their proposed idea of using visual prompting enhanced the VLM's reasoning capability and it had a significant impact in bridging the gap between a robot's visual perception and its manipulation task.

This paper [39] compares three leading detectors-YOLOv9, YOLOv10 and RT-DETR-on a bespoke set of farm images featuring sugar beet, monocot and dicot weeds, testing each model size from nano to large and resizing inputs between 320 and 960 px. Authors train the networks in the Ultralytics pipeline with heavy augmentations for 250 epochs, then record mean Average Precision (mAP) alongside run times on both GPU and CPU. Findings reveal that the YOLO family-especially the newest v9 and v10-strike the best balance, scoring high mAP while processing frames much quicker than RT-DETR. Although RT-DETR still excels on fine or confused weed classes, its multi-head attention and deep decoder layers push up latency and memory demand. YOLO wins here mainly because its compact CNN, swift PGI and GELAN fusions, plus NMS-free setup, yield far lower FLOPs, fewer parameters and much faster inferences, making it better suited to budget-limited, real-time farm robots.

The authors [33] of the paper introduced the PiVOT(Prompting with Iterative Visual Optimization) method which is a visual prompting based mechanism to enhance generic visual object tracking(GOT). Traditional visual object tracking systems usually struggle in various conditions like changing lighting, occlusions and the presence of distracting objects. The main goal of PiVOT is generating visual prompts in order to help the object tracker to identify and focus on relevant objects while suppressing distractors. This goal is achieved through a Prompt Generation Network (PGN) and Test-Time Prompt Refinement (TPR). First of all, the researchers used the PGN that took the features of the current and reference frame and it generated an initial score map which highlighted the potential target locations based on the current frame and reference frames. After that, pretrained Visual Language Model(CLIP) was used for the refinement of the prompts by evaluating similarities between candidate objects and reference templates. Further, the Relation Modeling (RM) module processed this visual prompt so that the tracker could stop focusing on distracting objects. The model was trained with a frozen ViT-L backbone(from DINOv2) to extract features efficiently. PiVOT demonstrated better performance across several benchmarks like NfS, LaSOT, AVisT, GOT-10k, and TrackingNet which showed

that the method could track the target object better even in challenging scenarios like occlusions or object deformation. Additionally, after comparing with other baseline trackers like ToMP or SiamRPN++ , it was showed that PiVOT achieved higher precision because of having the ability to dynamically refine prompts using the knowledge from CLIP. Even though this method improved the object tracking accuracy, the inference time got increased due to using foundational models like CLIP and DINOv2. Despite these challenges, the research showed that visual prompting can enhance the object tracking performance.

Jiang et al. (2024) [37]introduced T-Rex2 which is a novel approach for open-set object detection by combining textual and visual prompts to improve the generalization and robustness of object detection models. Traditional object detection models usually struggle to detect unseen objects due to working in a closed frame environment. Previous studies have shown strong performance in object detection by leveraging zero shot detection of CLIP and BERT, but text based methods struggle when in case of complex objects which are difficult to describe. On the other hand, visual prompts such as bounding boxes and points had shown a significant improvement in object detection but these prompts were less effective in capturing abstract concepts. Hence, the authors proposed the idea of T-Rex2 which addressed these limitations by integrating both text and visual prompts within a single model. The model was based on the DETR architecture and it incorporated two parallel text and visual prompt encoders to process multimodal inputs. T-Rex2 demonstrated strong zero shot object detection capabilities after evaluating on several existing benchmarks including COCO, LVIS, ODinW and Roboflow100. The model achieved high performance in both common and rare categories of images where the text prompts performed better in detecting frequent objects and the visual prompts excelled in detecting rare or unusual categories. Additionally, T-Rex2 outperformed existing models like GLIP and Grounding DINO in various test cases, specially when tested in zero-shot settings where the model was not trained on the specific evaluation datasets. One limitation of this research is that in case of common objects, the method faced difficulty in aligning text and visual prompts. Nevertheless, this research demonstrated high performance in detecting objects in dynamic environments by utilizing both text and visual prompting.

In this paper[47] the author designs a VP method that optimizes the visual prompts uniformly across all patches and introduces inductive biases between the rows and columns of the image patches. Previous VP techniques either affected a certain area or lost the info due to resizing. So they proposed a novel method LOR-VP[47] that leverages low-rank matrix multiplication to efficiently manage visual prompts. And their method performs better then previous SOTA method, AutoVP[28] by an average of 3.1% on seven network less training times. Using Auto vp and ILM-VP, Auto-VP and LP , they generate the prompt parameter in low rank using low rank multiplication rather than the full matrix and applied on image pixels. The low rank means shearing information through rows and columns. First the split the image into patches then LOR-vp prompt is added then send it through VIT and out put class label (some pretrain version Vit-B/32, Vit-B/16). They used VIT (CLS)to-

kens for classification. Which is more efficient than previous models because prompt optimizers were applied uniformly without structural consideration causing inconsistency. They choose the ImageNet-21K benchmark dataset for pretrain. And test effectiveness and efficiency used ImageNet-1K, Tiny-ImageNet and CIFAR-10/100. The performance on these datasets increased significantly. With CIFAR-10 the accuracy was 97.8% with ViT-B/16 and 96.98% with ViT-B/32. Similarly with CIFAR-100, LoR-VP achieved 89.52% with ViT-B/16 and 88.48% with ViT-B/32, outperforming AutoVP and other baselines

2.3 Summary of Key Findings

From the above discussion, it is observed that autonomous systems have seen huge advancements in the past few years by the help of deep learning techniques. The fine-tuning of pretrained models demonstrated remarkable achievements in this sector. However, the existing models still struggle in detecting objects under diverse weather conditions. Various strategies have been utilized to improve the detection accuracy which increase the computational complexity. In recent years, many researchers have used a technique named visual prompting to adapt pretrained models using a small number of parameters. The technique has been largely adopted for classification tasks. A few research has addressed this technique for object detection task where spatial distortion is present. So, the integration of visual prompting for object detection task remains an underexplored area. So, the visual prompting based approach with object detection models need to be analyzed for sccenarios where global distortion is present.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Our study used a deep learning based pretrained model YOLOv8 as a base object detection model. Moreover, we implemented a modified model with the ResNet50 backbone. We used a learnable visual prompting technique for both models. This technique was incorporated with both the models. After that, the models were trained and tested across five weather conditions. In order to train the models, we needed a high-end GPU and Pytorch library.

3.2 Societal Impact

The social impact of our study is to enhance the navigation, increase safety and reliability of autonomous systems. The major sectors are transportation, healthcare and agriculture. Improving object detection in adverse weather can reduce accidents and increase the adaptability of autonomous systems in daily life. In healthcare and agriculture our study could accommodate more dependable navigation capacity to assist people in need. It introduces VP using in this particular sector hence opening doors for future researchers to take initiation to work.

3.3 Environmental Impact

Our method uses less parameters for training which keeps the system as a very light weight design. Specially, our modified YOLOv8 with ResNet backbone, minimize the computational power rather than traditional heavy models. Less computational power makes our method more efficient for the object detection system , reducing environmental footprint by using less energy. Our thesis supports green computational goals and promotes long term environmental benefits.

3.4 Ethical Issues

Data protection, prejudice reduction, and the proper application of AI are all ethical issues. Our research consists of cleaning and managing dataset, mitigating biasedness, algorithm fairness, maintaining transparency and using proper measures to determine the result. It is essential to adapt a trustable decision making system so that the user can rely on it in navigating context. Considering autonomous systems have an impact on human safety, it is a professional obligation to provide open reporting and ongoing validation before practical deployment.

3.5 Project Management Plan

Our thesis was a structured plan divided into several phases. First of all, we planned a Literature Review to identify the research gaps and relevant models. Secondly, we collected the data set and started Preprocessing. Prepared datasets under various weather conditions. Model Integration was our next step. We tried different models and chose to implement VP with YOLOv8 and tested baseline performance. After that we studied about making it more efficient. Upon trial we decided to implement Resnet50 as the backbone for our base model. Next was the evaluation, where we conducted experiments, used new techniques of VP that restore the image and trained the model with less parameters. Then we documented the results. At the end we compiled the results and the final paper. As resources we used GPU-enabled system, publicly available datasets and open sources libraries.

3.6 Risk Management

The project has a high cost-benefit ratio because of using open-source frameworks and pre-trained models to generate high research value with minimal financial commitment. By increasing perception reliability, VP integration in autonomous systems can ultimately lower maintenance costs. In the business world, the method provides scalability to companies creating intelligent transportation systems, smart surveillance, and driverless cars. Thus, the study contributes significantly and affordably to safer and more efficient automation.

3.7 Economic Analysis

The project has a high cost-benefit ratio since it uses free frameworks and pre-trained models. That reduces financial input while providing high research value. In the long run, adopting VP into autonomous systems can lower maintenance and accident costs by boosting perception reliability. Economically, the method allows adaptability for companies creating intelligent transportation systems, smart surveillance, and driverless cars. Thereby, the study makes a significant and affordable contribution to automation which is safer and more effective.

Chapter 4

Methodology

The research aims to enhance the object detection process by integrating visual prompting along with the standard fine-tuning process of object detection models under diverse weather conditions.

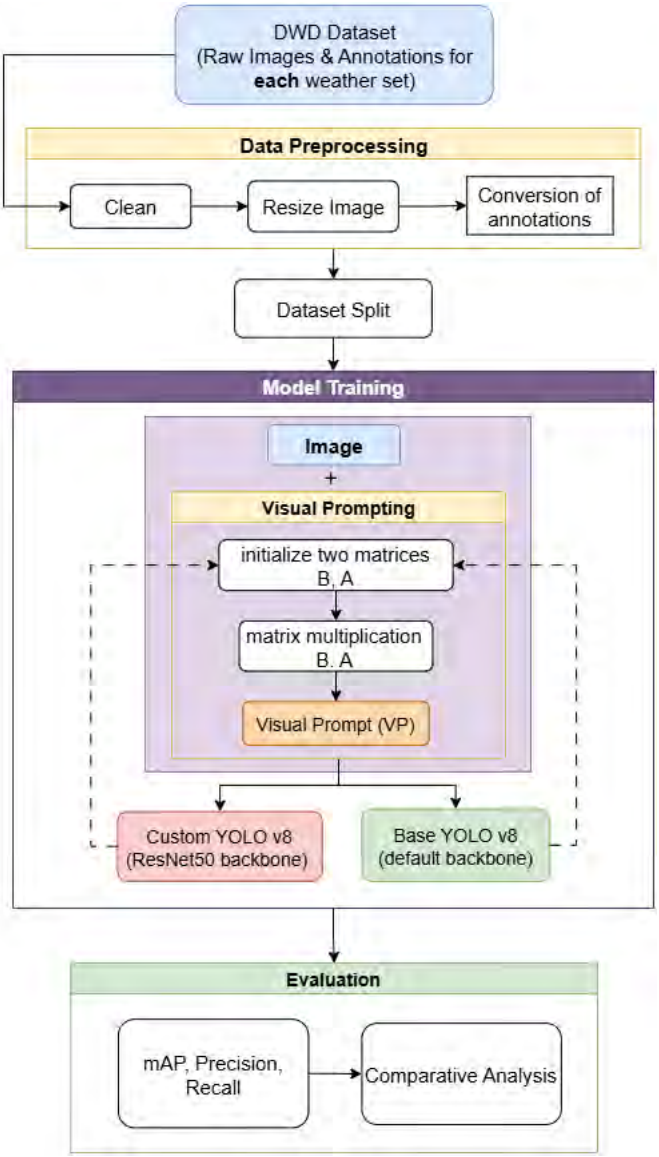


Figure 4.1: Top-level overview of the proposed system

Figure 4.1 provides a high level view of the proposed methodology. This research includes some main stages. To begin with, road image data in diverse weather and lighting conditions along with proper annotations need to be collected for training and testing purposes. After that, the data needs to be pre-processed in order to prepare for model implementation. During the pre-processing stage, various preprocessing techniques including data cleaning, resizing and preparing annotation files will be performed to prepare the data for using training the models. Further, each weather data will be splitted into train, validation and test set in order to train and test the models. After that, the proposed method needs to be implemented. So, the required environment needs to be created. After that, the visual prompt will be designed and it will be integrated with the YOLOv8 model. Due to this integration, the prompt will be able to learn from the standard loss function of the model and get optimized during the training. Further, a modified YOLOv8 model with a ResNet50 backbone will be implemented to analyze the performance of visual prompting on the ResNet50 backbone. Then the same prompt will be integrated with that model. After that, the models will be trained using specific hyperparameter settings. Then the performance of our proposed method using different matrics such as mAP50, mAP50-95, precision, recall. These results will be compared to the performance of the base YOLOv8 model and the modified YOLOv8 model with the ResNet50 model.

4.1 Dataset

In order to start our work, we wanted to find a good dataset so that our model can learn to detect objects more efficiently. For our thesis, we wanted to evaluate the performance of our model across different weather in different lighting conditions. Our primary goal was collecting road data with different objects in diverse weather. Collecting and labelling large datasets in different weather conditions would be difficult and time consuming for us. Moreover, maintaining good quality data was indeed a great challenge for our task. As a result, we decided to utilize publicly available dataset in order to train and test our model.

4.1.1 Dataset Description

We have used a benchmark dataset named "DWD(Diverse Weather Dataset)" for our research. This dataset consists of five sets where each of the sets contains road scenarios of different weather conditions like daytime-clear, daytime-foggy, dusk-rainy, night-clear, and night-rainy. The daytime-clear subset refers to the images which are taken during daytime with moderate light and so the image quality is not degraded. The daytime-foggy weather condition set contains foggy weather images that are taken during daytime and so the image quality mainly degrades due to the effect of fog. After that, the dusk-rainy weather condition imageset consists of images that are taken during the rainy weather condition in lowlight. The night-clear imageset contains images of night scenarios where the weather was normal. Lastly, the night-rainy scenario was the most challenging scenario as the images are rainy scene images taken during the nighttime. This dataset is created using three different dataset Cityscapes [8], Berkeley Deep Drive (BDD-100K) [9] and Adverse Weather [2]. In this dataset, each set consists of two folders where one is the image folder and

the other one is the annotation folder. The image folder consists of images of road conditions in a particular weather condition. The annotation folder consists of the object class name and the bounding box annotation in PASCAL VOC (.xml) format. The daytime-clear, night-clear, night-rainy, dusk-rainy, daytime-foggy image folders consist of 27708 images, 26,158 images, 2494 images, 3501 images and 3775 images respectively. In these images, there are objects of seven predefined categories which are Bus, Bike, Car, Motor, Person, Rider, Truck.

Day-Clear Night-Clear Night-Rainy Day-Foggy Dusk Rainy

Figure 4.2: Few images of different weather conditions from DWD (Diverse Weather Dataset)

4.1.2 Data Preprocessing

Data preprocessing is one of the most important steps in preparing a dataset for training a model. For our specific dataset, we had to clean the dataset, then resize the images to a specific size and prepare the annotation files to train and test the YOLOv8 model.

To begin with, we first checked whether each image file has its corresponding annotation file. We did not find any such image in daytime-clear, daytime-foggy, dusk-rainy and night-rainy folders. However, we found 8996 images in the night-clear folder that did not have annotation files. So, we dropped those files as manually labelling those images would become time consuming. Moreover, the night-clear folder already had 26,158 images and we still had a lot of images for training the model. After that, we checked whether we have any corrupted image and annotation files in each set of the dataset. There were no such images and annotation files in this dataset. Further, we checked whether each image folder has a valid annotation file or not. We did not find this kind of file in daytime-clear weather. We found daytime-foggy, night-clear, dusk-rainy and night-rainy folders have 19, 184, 16 and 19 empty annotation files respectively. So, we decided to drop the image and annotation files. Since we are working with a large dataset, dropping these images did not create a huge problem. Moreover, we checked whether any duplicate file was present in the dataset. However, we did not find any file like that. As a result, we did not have to handle the duplicate file.

Moreover, the images present in the folders were of 1280x720 pixels. However, for using lightweight object detection models, it was important to resize the images to square size since most of them use the same width and height for each image. Along with that, for handling huge data, we needed a certain size of image so that the training process can be faster too. So, we decided to resize the image into 640X640 pixels for each image. In this way, we could efficiently use the images for training.

After that, we had to prepare the annotations correctly for training the model. As mentioned before, the dataset contains annotations in PASCAL VOC format(.xml) which is not appropiate for training YOLO models. So, we had to convert it into YOLO format (.txt) as this format is required to train YOLO models. So, first of all, we have used class id for each class present in the dataset instead of class names. Then we calculated the middle point of each object class in x and y coordinates since in the XML annotation, the coordinates are measured for the corner values of each coordinate. Along with that, we had to normalize the bounding box values in the range of 0 to 1 so that the model can process the bounding box values. After that, we manually checked the bounding box values after mapping onto a few images. Thus we prepared our dataset in a way so that we could train our model.

Day-Clear Night-Clear Night-Rainy Day-Foggy Dusk-Rainy

Figure 4.3: Few annotated images of different weather conditions from DWD (Diverse Weather Dataset)

4.1.3 Data Splitting

Each folder of the dataset was splitted into 70% train set, 10% validation set and 20% test set.

Weather Condition	Train	Validation	Test
Daytime-Clear	19,324	2,760	5,523
Night-Rainy	1,732	247	496
Night-Clear	11,884	1,697	3,397
Daytime-Foggy	2,629	375	752
Dusk-Rainy	2,439	348	698

Table 4.1: Split Distribution of the Dataset

4.2 Proposed Method

Our proposed system involves designing learnable visual prompt and integrating the prompt with the YOLOv8 model and the customized YOLOv8 model with ResNet50 backbone. Due to this integration, the visual prompt will get updated during training time and enhance the performance of the models.

4.2.1 Visual Prompt Integration

Our proposed system involves integrating visual prompt with the input image so that the prompt can get updated during the training time.

Prompt Construction:

In order to implement the visual prompting mechanism, the first step is to resize the image. During the data pre-processing step, the image was resized to a size $L \times L$ where L represents the height or width of the image. Various visual prompting methods usually resize images to a size less than the default image size L. This strategy is usually used to add visual prompts around the image. Due to using this resizing strategy, important information of the images gets lost. To solve this problem, the image resolution was retained at this stage of our method.

In diverse weather scenarios, different weather conditions affect the overall image. As a result, a visual prompt that covers the entire image is necessary. However, a prompt with resolution $L \times L$ will have a total parameter count of cL^2. This will significantly increase the total number of parameters in the base model. To address the issue, we have implemented a strategy using two low rank matrices which was inspired from the paper [47].

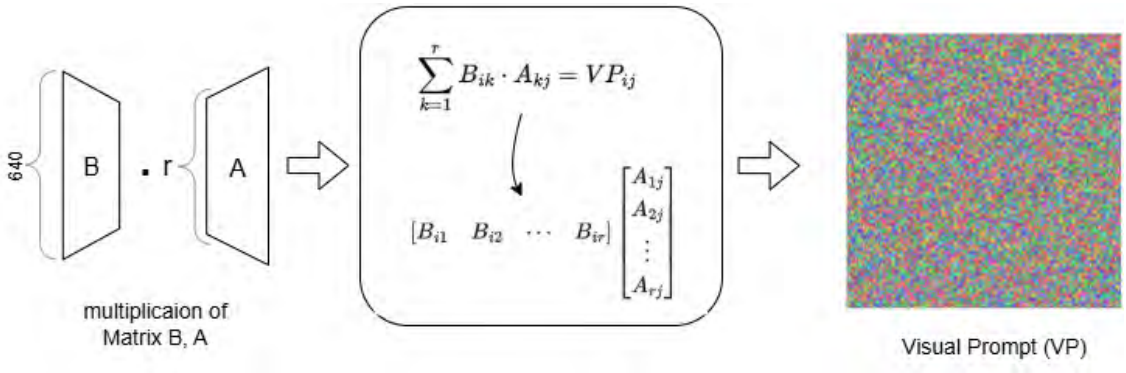


Figure 4.4: Construction of the Visual Prompt

First of all, two tunable matrices $B \in \mathbb{R}^{c \times L \times r}$ and $A \in \mathbb{R}^{c \times r \times L}$ are initialized. Here,

c = color channel
r = rank of the matrices
L = height or width of the image
and r $\ll$L

In this research, we have used RGB images and so the number of color channels is 3, we have used rank, r = 4 « L and we have chosen image height and width = 640 since the standard image size for the object detection model is 640.
The product of these matrices, $B \cdot A$, serves as the visual prompt for this image. So, the prompt, VP can be represented as following equation 4.1 :

$$\text{VP} = B \cdot A \in \mathbb{R}^{c \times L \times L} \tag{4.1}$$

After that, this visual prompt was added to the resized input image. So, the prompted image can be expressed as equation 4.2 :

$$P(x) = x + VP; x \in D \tag{4.2}$$

Here, x represents an image from the Dataset, D.
Thus the visual prompt is constructed and added to the images present in the dataset.

Parameter Count

The order of the matrix B is c ×L ×r and the order of the matrix A is c ×r ×L. So, the parameter count for this visual prompt design,

$$C = c \times L \times r + c \times r \times L = 2 \times c \times L \times r \tag{4.3}$$

The visual prompt could be initialized as a random matrix of size $L \times L$, but then the parameter count would become $c \times L \times L$. Thus, the number of tunable parameters has been reduced from cL^2 to $2cLr$. As we have used c = 3, L = 640 and r = 4, the total parameter count for our design becomes 15360.

4.2.2 Detection Models and Prompt Optimization

After constructing the visual prompt, the prompt needs to be integrated with the models. In this research, the YOLOv8 model and a modified YOLOv8 model with a ResNet50 backbone is used where the visual prompt was integrated.

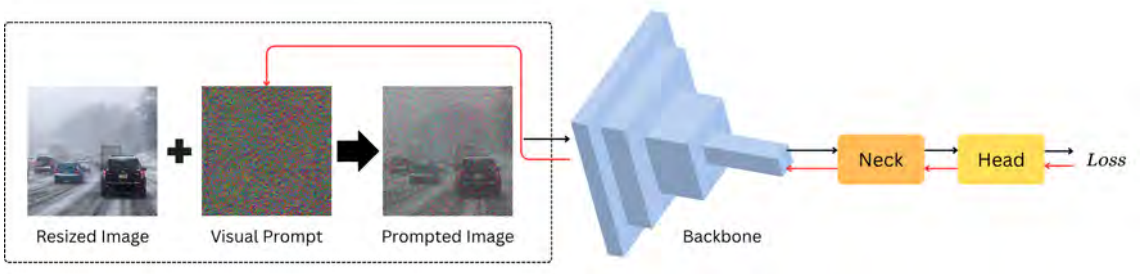


Figure 4.5: Architecture of the proposed system

YOLOv8 Model

The You Only Look Once version 8 (YOLOv8) model [44] was used in this research as the base object detection model. The model was pretrained for object detection tasks using the COCO dataset, which gave the backbone and neck a strong feature representation capabilities before fine-tuning. This model is widely known for its detection accuracy while maintaining real-time performance. Due to the lightweight design of the model and efficiency in object detection, the model has been utilized in this research.

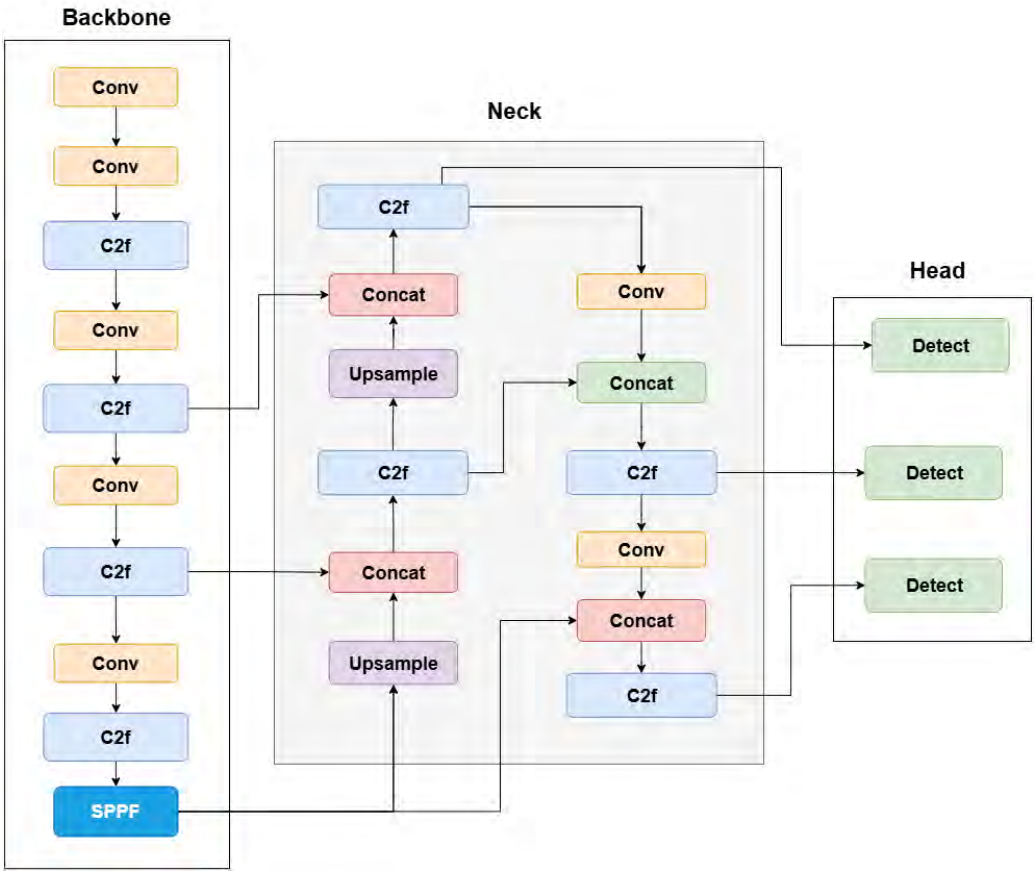


Figure 3.6: Architecture of the YOLOv8 Model

The YOLOv8 model consists of three core components which includes backbone, neck, and head as shown in Fig. 3.6. The prompted image goes to the backbone which extracts features from the input image. YOLOv8 employs an advanced Cross Stage Partial (Cross Stage Partial modules) based backbone. This backbone is optimized for both speed and accuracy. The extracted features are then refined by the neck module of the model. The model utilizes an optimized Path Aggregation Network (PANet) in order to enhance the flow of information throughout different feature levels. After that, the head provides the final prediction from the refined features. The final prediction includes the coordinates of the bounding box, confidence score of the objects, and the class labels. YOLOv8 introduces an anchor free approach to bounding box prediction instead of the anchor based predictions present in the previous versions of the model. This approach reduces the number of hyperparameters and enhances the flexibility of the model in handling objects of different ratios and scales.

After the prediction is done by head, the loss function is calculated. The model parameters are updated based on the calculated loss. During the same time, the prompt parameters are jointly optimized.

Customized YOLOv8 Model with ResNet50 Backbone

In this research, we have used a modified YOLOv8 model with a ResNet50 Backbone[6] as shown in Fig.3.7. ResNet50 was originally trained using the ImageNet-

1K dataset for classification tasks. This is widely used as a backbone of various object detection frameworks like Faster-RCNN models because of its feature extraction and generalization capabilities. We have integrated this Resnet50 model with the lightweight YOLOv8 model as a backbone to analyze the performance of this modified model under diverse weather scenarios. Moreover, we wanted to evaluate whether visual prompting can increase the efficiency of the detection process under diverse weather across different backbone architectures. Thus we wanted to compare the performance of proposed visual prompting in the object detection process across backbones that are optimized for classification and detection.

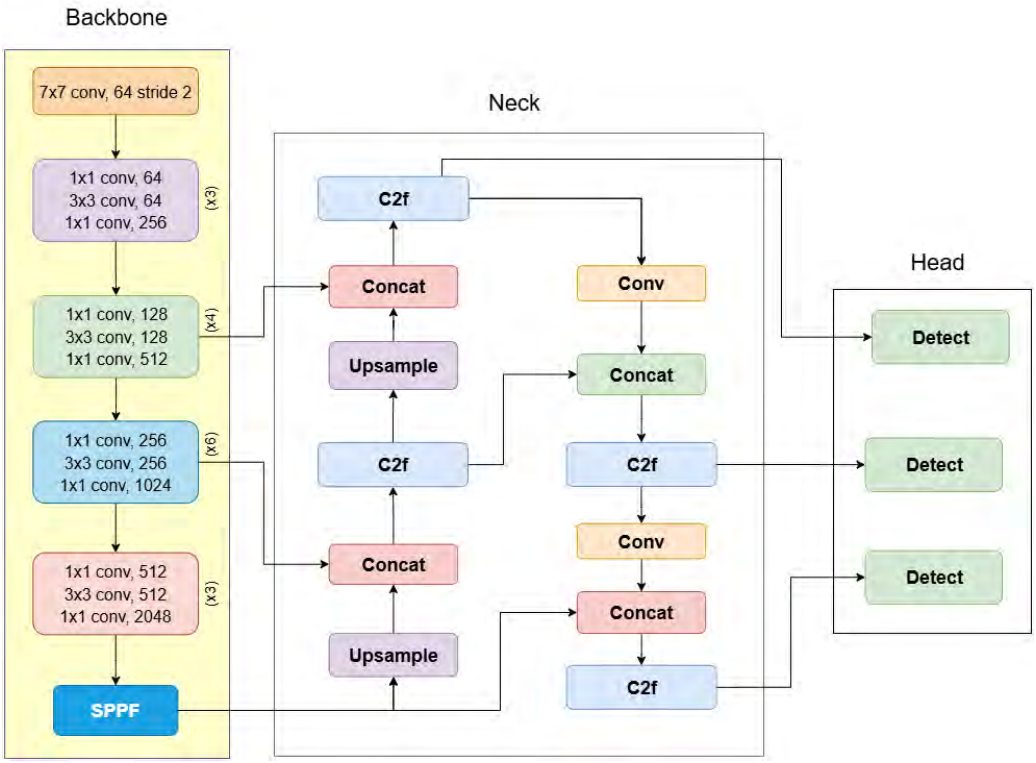


Figure 3.7: Customized YOLOv8 with ResNet50 Backbone

The prompted image goes to the ResNet50 based backbone which is responsible for feature extraction. The ResNet50 model consists of 50 layers where the first layer is a 7 X 7 convolutional layer with a stride 2. There is a 3 X 3 max pooling layer with stride 2 after the first layer. The further layers consist of ResNet Layers where each layer has multiple ResNet Blocks. Each ResNet block consists of 3 layers where each layer has 1 X 1, 3X3 and 1X1 convolutions with skip connections. These connections are used to solve the vanishing gradient problem. Instead of the global Average Pooling Layer and the fully connected layer of the ResNet50 classifier, the Spatial Pyramid Pooling Fast (SPPF) module is used for multi-scale feature extraction. The refinement of the extracted features are done by the neck component. After that, the head predicts the bounding boxes along with class labels and confidence score. The default neck and head component of YOLOv8 model was preserved in the modified architecture.

After that, the loss function is calculated and during the backpropagation the prompt parameters get updated along with the model weights.

Chapter 5

Implementation Details and Result Analysis

5.1 Implementation Details

5.1.1 Requirements

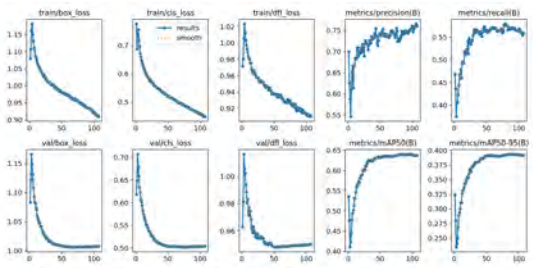
For this research, high-end GPU support was required as we had to handle a large volume of data efficiently. In order to enable GPU acceleration, an NVIDIA GeForce RTX 3080 Ti was used to conduct all the experiments. The models were trained in a Python 3.11 environment where Pytorch 2.7.1 with CUDA 11.8 support was utilized along with the corresponding torchvision library.

5.1.2 Model Training

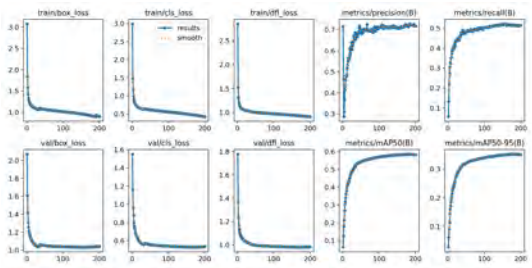
For each weather condition, four models were trained. Considering the GPU constraints and accuracy tradeoff, the YOLOv8-L model was used as the baseline method in this research which has a total 43,612,005 parameters. After that, the visual prompting module was implemented in order to integrate with the model. So, two matrices were initialized where the initial value of B and A was 0 and 0.01. This value was chosen so that the dot product of these matrices becomes 0. As a result, the visual prompt becomes 0 at first. This prompt module was added with the images before normalization. The matrices were added as parameters in the YOLOv8 model where the default AdamW optimizer was used with a learning rate of 0.02. So, the number of parameters became 43,627,365 after integrating the visual prompting with the YOLOv8 model. After that, the ResNet50 backbone was integrated with the neck of the YOLOv8 model. The total parameters of this model was 30,977,381. After transferring the pretrained weight of ResNet50, the model was trained for all weather conditions. Further, the visual prompt was added with this modified model in the same way and the model was trained. So, the total parameters became 30,992,741. For all scenarios, we have trained for 200 epochs with a patience value 20. As a result, the model could be trained on the same dataset for 200 times while using early stopping value 20 which means if the model does not see improvement for 20 epochs, the training will be stopped. Due to the computational requirements, we have used batch size 16 for each case. The default learning rate 0.01 and optimizer AdamW was used for training the YOLOv8 model and the mod-

ified YOLOv8 model with the ResNet50 backbone. Thus the models were trained in this research. The following figures show the training and validation performance of our proposed method across different weather conditions.

Daytime-Clear Weather: The following figures represent the training and validation performances of our proposed visual prompting integrated models.



YOLOv8 with VP under Daytime-Clear Weather

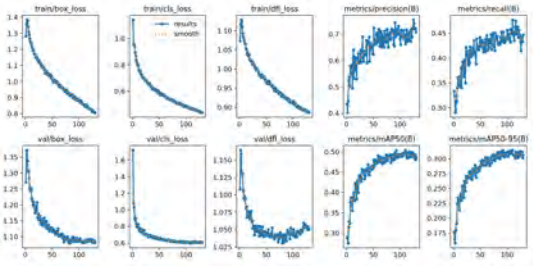


Modified YOLOv8 with VP under Daytime-Clear Weather

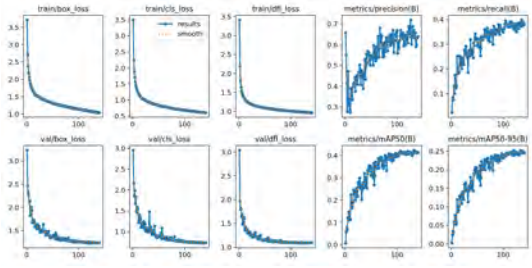
Figure 5.1: Training and validation performance for the proposed models under Daytime-Clear weather conditions

Fig. 5.1 represents the training and validation performance of our proposed visual prompting integrated models. There are three types of losses present including box loss, classification loss and Distributed Focal Loss. During the training time of the YOLOv8 model, these losses increased in the initial few epochs and started decreasing in a stable way. The precision, recall, mAP50, mAP50-95 followed the same trend. During the validation time, the same pattern was seen for box and class loss, but the dfl loss again increased a little during the last few epochs. The model was trained for 107 epochs as the mAP50-95 did not improve during the last 20 epochs. During the training and validation of the modified YOLOv8 model with visual prompting, all the losses decreased in a stable way and the performance metrics increased with time.

Daytime-Foggy Weather: The following figures represent the training and validation performances of our proposed visual prompting integrated models.



YOLOv8 with VP under Daytime-Foggy Weather



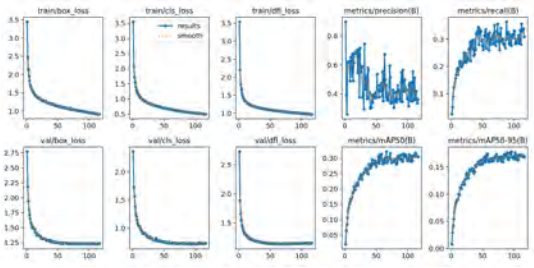
Modified YOLOv8 with VP under Daytime-Foggy Weather

Figure 5.2: Training and validation performance for the proposed models under Daytime-Foggy weather conditions

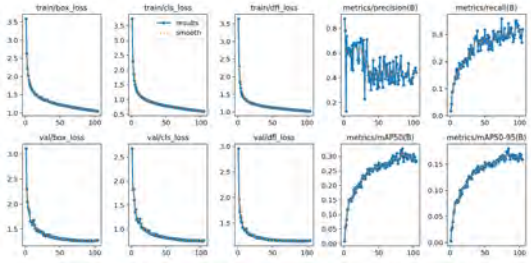
Fig. 5.2 represents the training and validation performance of our proposed visual

prompting integrated models. During the training time of the YOLOv8 model, all the losses decreased in a stable way. The precision, recall, mAP50, mAP50-95 increased with the time. During the validation time, the same pattern was seen for box and class loss, but the dfl loss again increased a little during the last few epochs. During the training and validation of the modified YOLOv8 model with visual prompting, all the losses decreased in a stable way and the performance metrics increased with time.

Dusk-Rainy Weather: The following figures represent the training and validation performances of our proposed visual prompting integrated models.



YOLOv8 with VP under Dusk-Rainy Weather

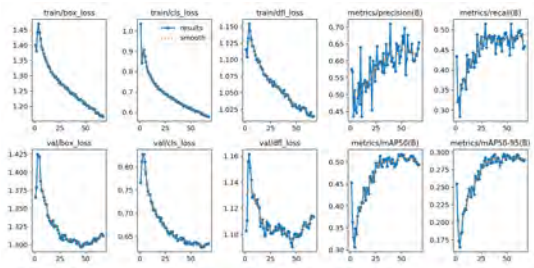


Modified YOLOv8 with VP under Dusk-Rainy Weather

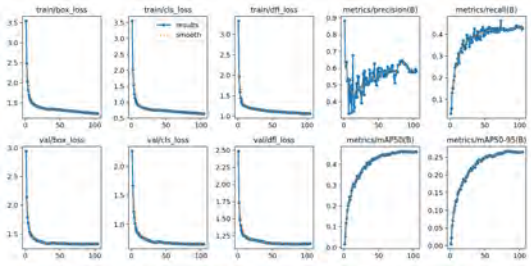
Figure 5.3: Training and validation performance for the proposed models under Dusk-Rainy weather conditions

Fig. 5.3 demonstrates the training and validation performance of the visual prompting integrated models under dusk-rainy weather conditions. All the losses decreased over the epochs during training and validation time. There were fluctuations present in the performance metrics. The model was trained till 116 epochs as the model stopped after not seeing improvement for 20 epochs. In the modified YOLOv8 model, the loss values decreased with time which shows the prediction of the model became better with time. The model was stopped at 104 epochs and the performance metrics were average due to the complex scenario.

Night-Clear Weather: The following figures represent the training and validation performances of our proposed visual prompting integrated models.



YOLOv8 with VP under Night-Clear Weather

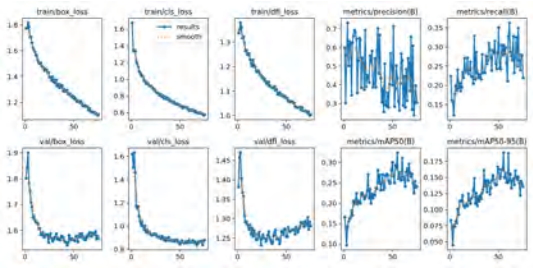


Modified YOLOv8 with VP under Night-Clear Weather

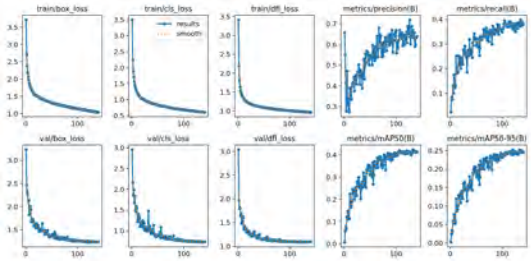
Figure 5.4: Training and validation performance for the proposed models under Night-Clear weather conditions

Fig. 5.4 shows the training and validation performance of our proposed visual prompting integrated models under night-clear weather. In this weather, the loss initially increased and then started decreasing. The model did not have to run for 200 epochs as the patience value triggered early stopping. During the validation time, the performance metrics overall increased while having fluctuations. The modified YOLOv8 model(ResNet50 backbone) showed a stable performance during training and validation. All the losses decreased over the epochs. The precision was initially 0.9 which decreased while the recall increased over the epochs. The mAP50 and mAP50-95 got improved which shows that the prediction of the model became better. Moreover, the model was trained for 104 epochs as the mAP50-95 did not improve for the last 20 epochs.

Night-Rainy Weather: The following figures represent the training and validation performances of our proposed visual prompting integrated models.



YOLOv8 with VP under Night-Rainy Weather



Modified YOLOv8 with VP under Night-Rainy Weather

Figure 5.5: Training and validation performance for the proposed models under Night-Rainy weather conditions

Fig. 5.5 represents the training and validation performance of our proposed visual prompting integrated models under night-rainy conditions. During the training of the YOLOv8 model, all the losses decreased steadily. The decreasing pattern was present in validation performance as well but with a little fluctuation. There was a huge fluctuation present in the precision metrics. Overall, the model could not show a strong detection accuracy on the validation set due to dealing with the complex night-rainy weather. For the modified YOLOv8 model, the losses also decreased where the losses dropped sharply during the first few epochs. Initially, there was a high precision value and a low recall value. During the training the precision decreased and the recall increased. The mAP50 and mAP50-95 got improved with time.

5.2 Result Analysis

5.2.1 Analysis Metrics

In order to evaluate the object detection perfromance, we have analyzed four metrics which includes precision, recall, mAP50 and mAP50-95.

Precision:

Precision refers to the proportion of correctly predicted positive samples (True Positives, TP) relative to all samples predicted as positive which includes True Positives(TP) and False Positives(FP). [1].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

Recall:

Recall measures the proportion of correctly detected positive samples (True Positives, TP) relative to all actual positives which includes all True Positives(TP) and False Negatives(FN)[1].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

Intersection Over Union(IoU):

Intersection over union measures the overlap bewteen the predicted bounding box and the ground truth bounding box. It is defined as the ratio between the overlap area and the area of union between these two bounding boxes[19].

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{5.3}$$

mean Average Precision(mAP):

Average Precision(AP) measures the area under the precision-recall (PR) curve. Mean Average Precision calculates the average of these AP values across different classes[19].

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{n} AP_c \tag{5.4}$$

where APc is the AP of class c and n is the number of classes.

Mean Average Precision at IoU threshold 0.5 (mAP50):

This evaluation metrics refers to the mAP calculated at the IoU threshold 0.5 which means if the 50% area of predicted bounding box and the ground truth bounding box overlapped, the detection is considered correct and the mAP50 is calculated using this[4].

Mean Average Precision across multiple IoU thresholds ranging from 0.5 to 0.95 (mAP50-95):

mAP50-95 refers to the average of the mean average precision that is calculated at different IoU thresholds. The range of these IoU threshold starts from 0.5 to 0.95. It provides an overview of model performance at different difficulty level of detection[4].

5.2.2 Performance Analysis

Daytime-Clear Condition:

In the clear weather scenario, Visual Prompting significantly boosted mAP50, mAP50:95, precision, recall in both scenarios. Integration of Visual Prompting along with the baseline YOLOv8 model improved mAP50 from 65.4% to 68.6%, precision from 67.7 % to 73.9 %, recall from 55.1 % to 58.4 % and mAP50-95 from 43.2 % to 45.7 %. In the modified YOLOv8 model, Visual Prompting improved mAP50 from 62.7 % to 64.7 %, mAP50-95 from 39.8 % to 42.6 % and recall from 51.0 % to 52.0 %. However, precision decreased slightly from 72.9 % to 72.0 % in this modified architecture.

Method	**Precision (%)**	**Recall (%)**	**mAP@0.5 (%)**	**mAP@0.5:0.95 (%)**
YOLOv8 w/o VP	67.7	55.1	65.4	43.2
YOLOv8 w/ VP	73.9	58.4	68.6	45.7
YOLOv8(ResNet50 backbone) w/o vp	72.9	51.0	62.7	39.8
YOLOv8(ResNet50 backbone) w/ vp	72.0	52.0	64.7	42.6

Table 5.1: Performance analysis of different methods under **Daytime-Clear** condition.

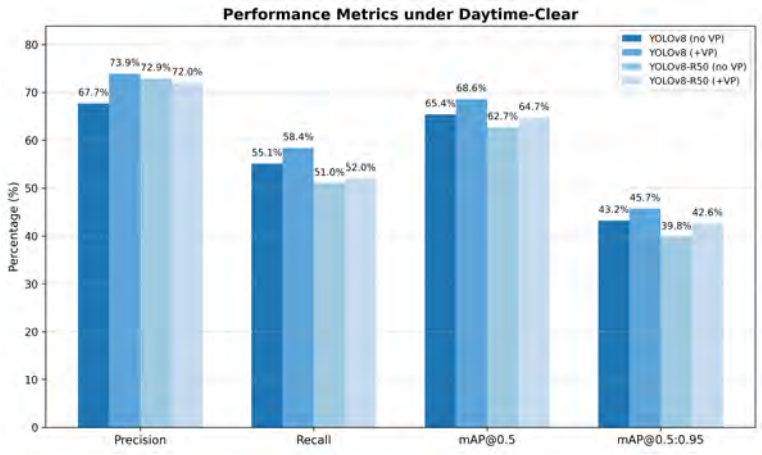


Figure 5.6: Performance analysis under **Daytime-Clear** weather condition

Daytime-Foggy Condition:

Visual Prompting enhanced the performance of YOLOv8 model in this scenario while increasing the precision from 64.0% to 68.8%, recall from 41.5% to 45.5%, mAP50 from 55.5 % to 60.0% and mAP50:95 from 37.4% to 41.8%. For the customized YOLOv8 model with ResNet50 backbone, mAP50 improved from 50.1% to 54.5%, and mAP50-95 rose from 0.323 to 0.367 and precision increased from 64.0% to 64.7%. Recall, however, decreased a bit in this case where the recall value became 39.5% from 39.9%.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
YOLOv8 w/o VP	64.0	41.5	55.5	37.4
YOLOv8 w/ VP	68.8	45.5	60.0	41.8
YOLOv8(ResNet50 backbone) w/o vp	64.0	39.9	50.1	32.3
YOLOv8(ResNet50 backbone) w/ vp	64.7	39.5	54.5	36.7

Table 5.2: Performance analysis of different methods under **Daytime-Foggy** condition

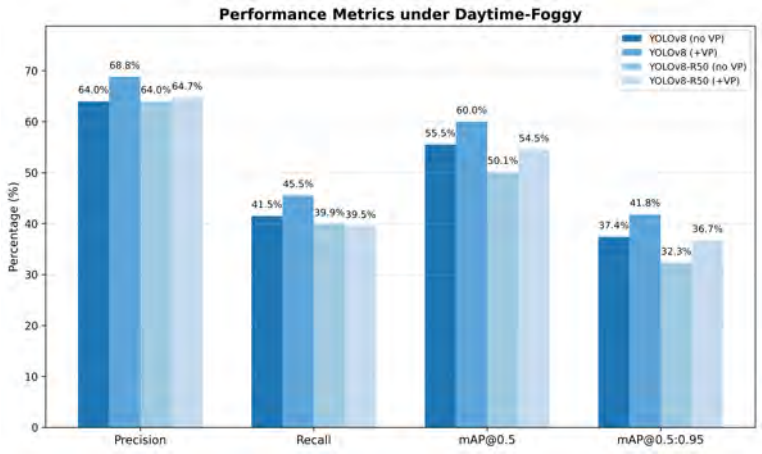


Figure 5.7: Performance analysis under **Daytime-Foggy** weather condition

Dusk-Rainy Condition:

In dusk-rainy weather condition, both of the models struggled in detecting objects. This is because both lighting and the rain degraded the quality of the images. In the YOLOv8 model, the mAP50 increased from 39.8% to 42.9 %, mAP50:95 from 25.2% to 28.4% and recall improved from 28.6% to 34.3%. However, precision decreased from 50.9% to 49.1%. On the other hand, for the ResNet50 backbone, Visual Prompting improved the mAP50, mAP50-95 and precision, but recall dropped slightly.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)
YOLOv8 w/o VP	50.9	28.6	39.8	25.2
YOLOv8 w/ VP	49.1	34.3	42.9	28.4
YOLOv8 (ResNet50) w/o VP	38.1	31.4	34.3	21.2
YOLOv8 (ResNet50) w/ VP	40.5	30.6	36.6	23.3

Table 5.3: Performance analysis of different methods under **Dusk-Rainy** condition.

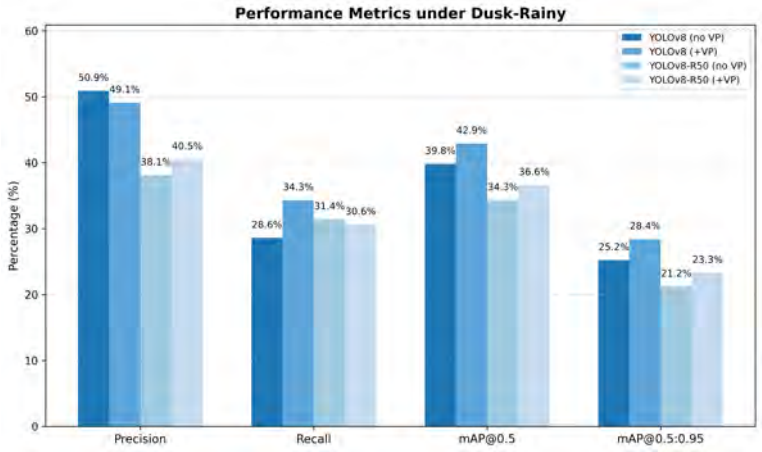


Figure 5.8: Performance analysis under **Dusk-Rainy** weather condition

Night-Clear Condition:
In this scenario, the performance of YOLOv8 model significally improved when Visual Prompt was integrated. For YOLOv8 model, mAP50 rose from 52.2% to 57.7%, mAP50-95 improved from 33.4% to 36.8%, precision increased from 61.7% to 63.7% and recall increased from 42.0% to 47.3%. Similarly, with the ResNet50 backbone, Visual Prompting showed marginal improvement in mAP50, precision and recall. The mAP50-95 was same in both cases.

Method	**Precision (%)**	**Recall (%)**	**mAP@0.5 (%)**	**mAP@0.5:0.95 (%)**
YOLOv8 w/o VP	61.7	42.0	52.8	33.4
YOLOv8 w/ VP	63.1	47.3	57.7	36.8
YOLOv8 (ResNet50 backbone) w/o VP	54.1	42.7	50.8	32.0
YOLOv8 (ResNet50 backbone) w/ VP	54.6	42.9	51.3	32.0

Table 5.4: Performance analysis of different methods under **Night-Clear** condition.

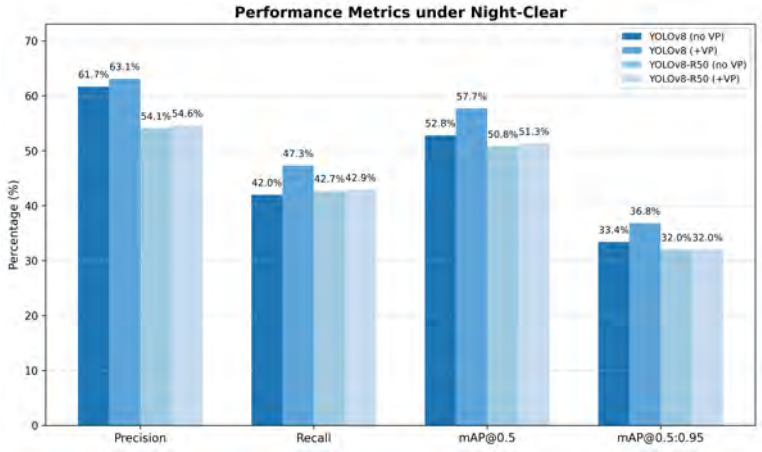


Figure 5.9: Performance analysis under **Night-Clear** weather condition

Night-Rainy Condition:

Under Night-Rainy condition, Visual Prompting showed significant improvement in both scenarios. For YOLOv8 model, integration of Visual Prompting improved mAP50 from 27.1% to 32.5%, mAP50-95 from 16.1% to 19.0%, recall from 22.0% to 31.6%. Similarly, with the ResNet50 backbone, Visual Prompting increased mAP50 from 28.6% to 32.2% and mAP50-95 rose from 16.0% to 18.6%. Nonetheless, precision decreased slightly by 1.4% and 0.3% respectively for both models after applying Visual Prompting.

Method	**Precision (%)**	**Recall (%)**	**mAP@0.5 (%)**	**mAP@0.5:0.95 (%)**
YOLOv8 w/o VP	40.8	22.0	27.1	16.1
YOLOv8 w/ VP	39.4	31.6	32.5	19.0
YOLOv8 (ResNet50 backbone) w/o VP	38.0	26.8	28.6	16.0
YOLOv8 (ResNet50 backbone) w/ VP	37.7	26.5	32.2	18.6

Table 5.5: Performance comparison of different methods under **Night-Rainy** condition.

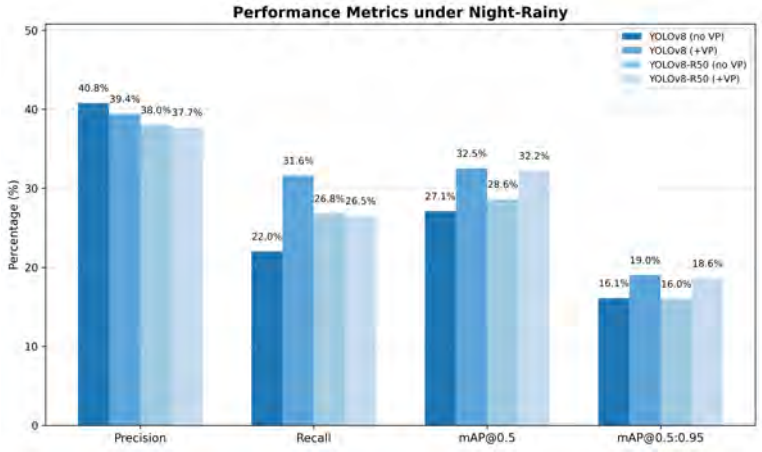


Figure 5.10: Performance analysis under **Night-Rainy** weather condition

5.2.3 Class-wise Performance Analysis

We have analyzed the performance of these models across each weather by using their class-wise precision, recall, mAP50, mAP50-95.

Daytime-Clear Condition:

The effect of Visual Prompting on the performance of YOLOv8 under daytime-clear condition is shown in table 5.6.

Class	YOLOv8 w/o VP				YOLOv8 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	66.3	56.9	66.1	54.4	70.5	60.0	69.0	56.5
Bike	58.9	47.4	58.0	32.1	66.9	54.5	63.5	35.8
Car	80.9	74.6	82.7	59.3	84.3	75.0	83.8	60.1
Motor	66.7	47.2	59.3	31.5	73.7	51.5	62.4	33.6
Person	71.0	54.9	66.7	38.0	76.6	56.4	69.7	40.8
Rider	65.8	46.5	60.2	35.8	76.2	51.8	66.1	39.9
Truck	64.3	58.3	64.6	51.6	69.0	59.9	65.9	53.0

Table 5.6: Class-wise Performance Analysis of YOLOv8 With and Without **Visual Prompting** Under Daytime-Clear condition

From the above data, it can be seen that Visual Prompting could improve all metrics for all classes while integrated with the base YOLOv8 model. For YOLOv8 model, the car class showed the highest result as it was the most represented class in training scenario. However,the small and rare classes present in the dataset were benefitted the most by integrating Visual Prompting. For instance, the mAP of rider got improved by 5.5%, while the precision and recall increased by 10.4% and 5.3% respectively. On the other hand, the car class could see an improvement of 1.1% in mAP50. The precision and recall for this class increased 3.4% and 0.4% respectively.

The class-wise performances for the modified YOLOv8 model with ResNet50 backbone are represented in the following 5.7.

Class	YOLOv8+ResNet50 w/o VP				YOLOv8+ResNet50 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	70.1	51.9	63.8	50.9	69.5	54.5	65.2	53.2
Bike	65.2	45.3	54.9	27.8	61.3	45.1	57.5	30.8
Car	83.5	72.1	81.9	56.9	83.2	72.6	81.9	58.8
Motor	73.0	42.1	56.3	27.9	70.6	44.2	58.5	30.8
Person	76.8	48.5	63.8	34.3	75.7	50.0	65.8	37.7
Rider	71.8	42.3	55.5	31.6	71.4	42.1	60.1	35.9
Truck	69.7	55.1	62.8	48.9	70.5	55.4	63.7	51.2

Table 5.7: Class-wise performance analysis of YOLOv8 (ResNet50) with and without**Visual Prompting** under Daytime-Clear condition

The modified model give overall less performance compared to the YOLOv8 model. However, for all classes, mAP50-95 got increased by incorporating Visual Prompppting with the modified architecturer which shows that this method helps to improve the detection accuracy. All the classes except car class got better mAP50 while car class had a stable mAP50 score 81.9%. However, the precision of all classes decreased slightly after integrating Visual Prompting. Moreover, the recall value was improved almost in all classes while decreasing the recall slightly for rider and car class. Overall, incorporating Visual Prompting with the model, demonstrated better detection accuracy for most of the classes.

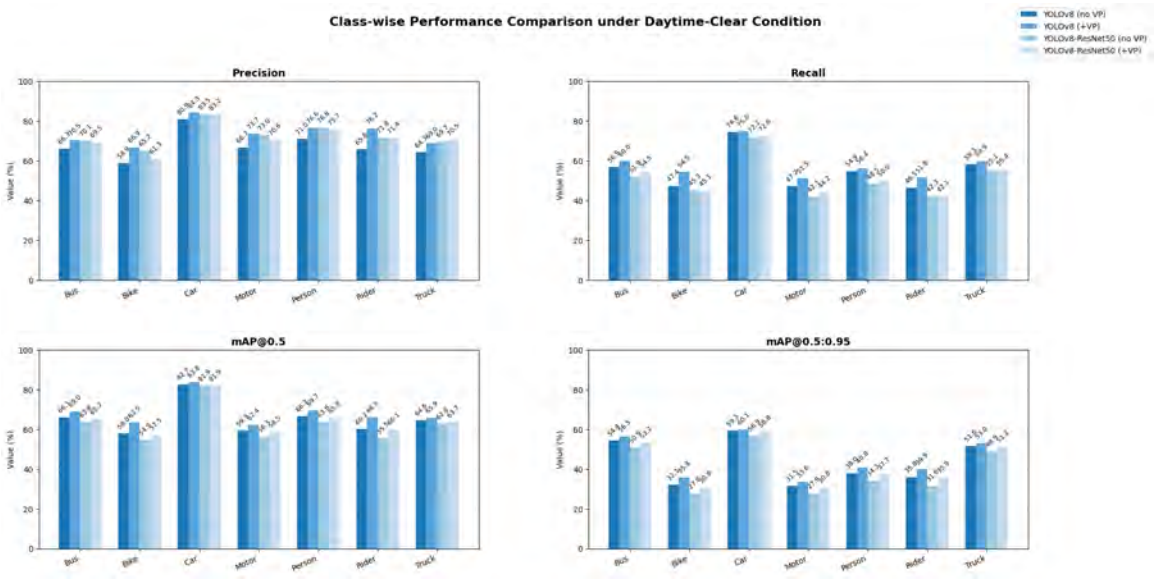


Figure 5.11: Class-wise Performance analysis under **Daytime-Clear** weather condition

Daytime-Foggy Condition:

In Daytime-Foggy condition, the accuracy decreased compared to the clear weather due to the degradation in image quality. The class-wise performance metrics of the base YOLOv8 model and the version integrating Visual Prompting are presented in the following table 5.8.

Class	YOLOv8 w/o VP				YOLOv8 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	68.9	37.2	55.1	42.8	71.6	35.0	55.8	43.7
Bike	56.9	34.9	46.8	28.0	61.0	41.1	52.5	32.0
Car	79.3	64.6	76.8	58.2	83.0	67.7	79.4	61.8
Motor	55.4	30.8	46.1	24.9	57.7	38.5	51.3	29.2
Person	71.9	41.7	60.1	37.2	78.2	46.3	64.9	42.7
Rider	71.2	47.3	63.0	40.9	76.0	53.6	68.0	46.7
Truck	44.3	33.8	40.9	30.0	54.1	36.0	47.7	36.4

Table 5.8: Class-wise Performance Analysis of YOLOv8 With and Without **Visual Prompting** Under Daytime-Foggy condition

After adding Visual Prompting, most of the classes demonstrated noticeable improvement. During daytime-foggy weather, the mAP50, mAP50-95 improved for all classes which shows that the detection accuracy was higher after incorporating Visual Prompting. The precision value for each classes also got improved which shows that the model could identify the objects more correctly after using Visual Prompting. The recall value of most classes also increased except Bus class where the recall slightly decreased.

Table 5.9 represents the performance metrics of the modified YOLOv8 model with ResNet50 backbone with Visual Prompting and without visual prompting.

Class	YOLOv8+ResNet50 w/o VP				YOLOv8+ResNet50 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	67.9	38.7	52.5	38.9	64.7	39.5	56.5	42.3
Bike	61.3	33.0	41.9	22.5	57.8	32.2	46.3	26.5
Car	78.5	61.3	72.8	53.0	76.9	62.6	74.9	56.3
Motor	53.3	28.6	34.9	19.4	54.9	27.5	42.0	24.4
Person	69.7	39.8	52.0	29.6	71.4	39.0	58.2	35.5
Rider	71.1	46.4	58.7	36.0	70.3	46.9	61.2	40.0
Truck	46.2	31.7	38.0	26.6	49.4	30.9	42.7	32.1

Table 5.9: Class-wise performance analysis of YOLOv8 (ResNet50) with and without **Visual Prompting** under Daytime-Foggy condition

For the YOLOv8 with ResNet50 backbone, adding VP contributed to significant improvements. In the scenario, the detection accracy got improved after using Visual Prompting technique. However, the effect of Visual Prompting was not same for the precision and recall values of all classes. For motor, person and truck class, the precision got improved, but the other classes see a decreament in precision. Similarly, it improved recall for Bus, Car and Rider while it decreased recall for the remaining classes.

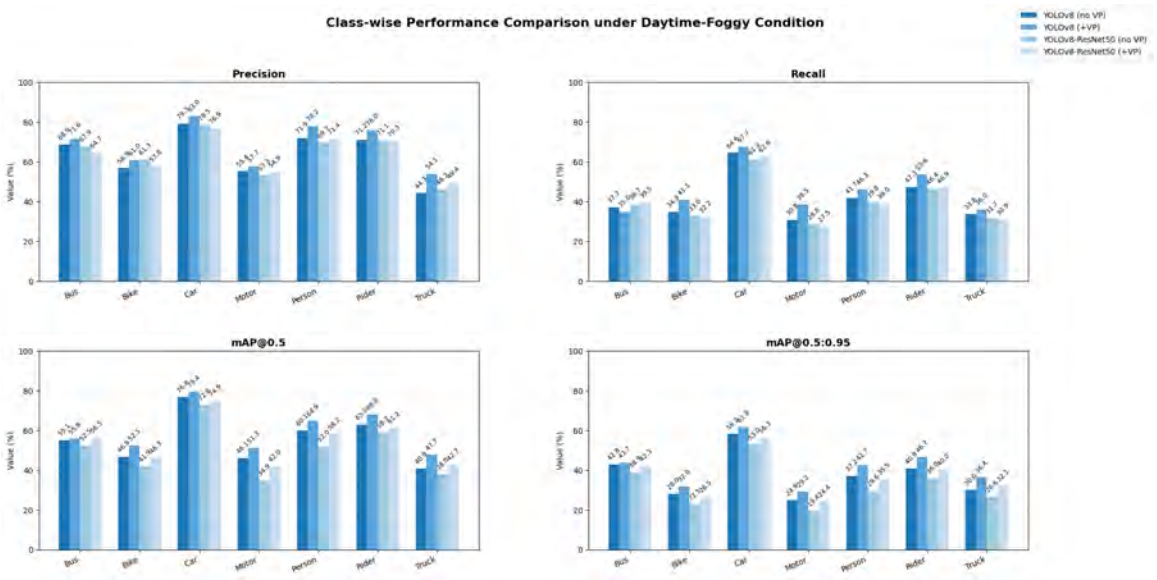


Figure 5.12: Class-wise Performance analysis under **Daytime-Foggy** weather condition

Dusk-Rainy Condition:

The impact of incorporating Visual Prompting in YOLOv8 model is presented in the following table5.10.

Class	YOLOv8 w/o VP				YOLOv8 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	47.7	32.2	39.9	31.7	53.2	43.9	48.7	37.4
Bike	50.0	9.0	30.2	18.0	29.4	16.9	26.9	17.2
Car	81.0	65.7	76.8	53.8	76.6	71.1	79.1	55.2
Motor	16.7	7.0	9.0	0.8	22.4	7.0	11.8	5.8
Person	68.6	32.6	51.5	24.9	71.2	37.6	56.3	29.0
Rider	37.5	8.8	21.1	10.6	36.4	11.8	22.4	13.7
Truck	55.2	43.4	49.9	36.8	54.6	50.9	55.1	40.3

Table 5.10: Class-wise Performance Analysis of YOLOv8 With and Without **Visual Prompting** Under Dusk-Rainy condition

In dusk-rainy weather condition, the performance of YOLOv8 model degrades due to the effect of low light and rain. In this scenario, integration of Visual Prompting resulted in significant gains for the YOLOv8 model. The detection accuracy increased significantly in most of the classes except Bike. For this class, both mAP50 and mAP50-95 decreased. In this scenario, the precision of Bus, Motor and Person class increased while the precision of other classes decreased. The recall value for most classes improved a lot in this scenario, which shows that adding Visual Prompting with the YOLOv8 model helps the model to reduce missed detections.

The class-wise performance analysis of Visual Prompting in the YOLOv8 model with ResNet50 backbone is represented in the following table 5.11.

Class	YOLOv8+ResNet50 w/o VP				YOLOv8+ResNet50 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	45.6	40.4	42.4	31.2	38.6	37.4	40.0	29.1
Bike	24.6	12.7	11.5	04.6	44.4	11.3	28.1	17.9
Car	73.2	67.0	74.9	49.6	72.1	66.9	75.5	51.3
Motor	06.0	07.0	10.7	04.9	09.0	07.0	08.0	05.5
Person	51.9	36.6	42.3	18.3	57.3	31.5	46.4	20.6
Rider	13.9	08.0	10.4	05.6	15.8	08.0	08.6	03.6
Truck	50.5	46.8	48.1	34.0	46.2	50.5	49.8	35.0

Table 5.11: Class-wise performance analysis of YOLOv8 (ResNet50) with and without **Visual Prompting** under Dusk-Rainy condition

From the mAP50 and mAP50-95 value, it can be seen that incorporating Visual Prompting improved the detection accuracy for most classes in the modified model. In this scenario, mAP50 and mAP50-95 significantly boosted for some classes like bikes, person. For most classes, the models maintained a balanced precision and

recall. Overall, in this scenario, due to the effect of two weather conditions including low-light and rain, the models struggled to detect objects correctly.

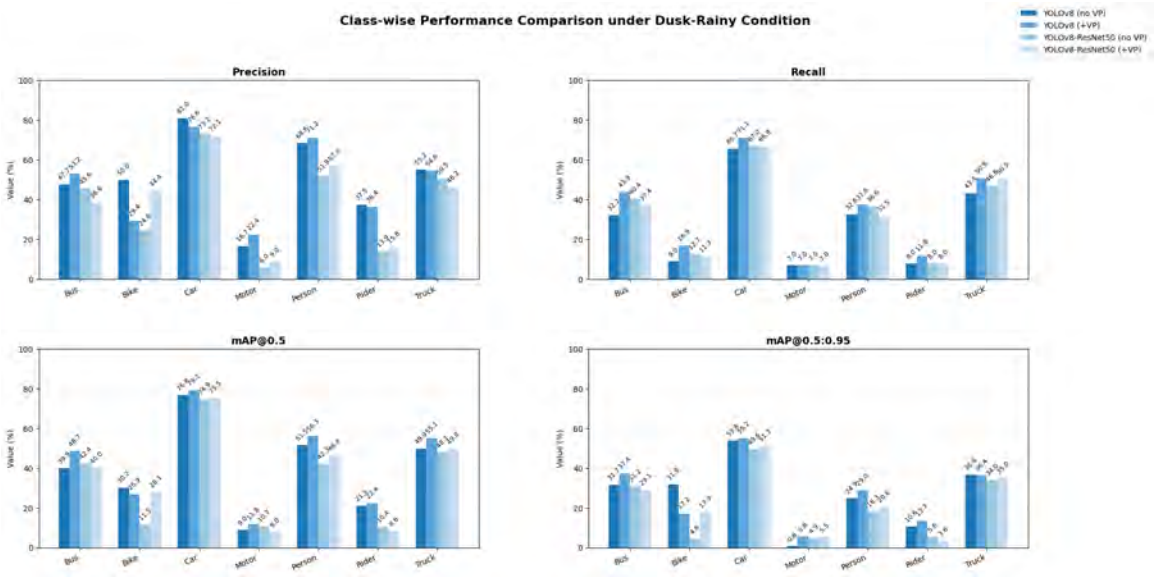


Figure 5.13: Class-wise Performance analysis under **Dusk-Rainy** weather condition

Night-Clear Condition:

The class-wise performances in the YOLOv8 model under night scenario are presented in the following table 5.12.

Class	YOLOv8 w/o VP				YOLOv8 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	57.9	52.3	56.7	47.0	59.8	55.9	60.0	50.1
Bike	45.9	25.4	33.9	14.8	56.9	36.6	46.1	22.6
Car	80.5	67.2	77.9	49.6	75.1	70.7	77.9	50.0
Motor	54.0	19.3	35.8	22.5	54.5	20.5	40.3	24.8
Person	71.8	45.7	59.6	30.7	66.6	52.9	63.1	34.0
Rider	64.2	33.1	49.2	26.4	72.1	39.5	58.3	31.1
Truck	57.8	50.8	56.2	42.7	56.4	54.8	58.0	44.7

Table 5.12: Class-wise Performance Analysis of YOLOv8 With and Without **Visual Prompting** Under Night-Clear condition

In this scenario, Visual Prompting could enhance the detection accuracy for the YOLOv8 model for all classes except the car class. After incorporating Visual Prompting, the mAP50 of the car class remained the same, but mAP50-95 got increased which shows that the model could perform better in different levels of detection difficulty. In most classes, integration of Visual Prompting along with the fine tuning of YOLOv8 affected positively in precision. The precision of Bus, Bike, Motor, Rider got improved while the recall value of all classes improved by using Visual Prompting. Thus the model could provide a stable result while incorporating

Visual Prompting.

This table 5.13 shows the effect of Visual Prompting on the performances of all classes using the YOLOv8 model with ResNet50 backbone under Night-Clear Scenario.

Class	YOLOv8+ResNet50 w/o VP				YOLOv8+ResNet50 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	52.6	54.7	56.6	46.6	51.3	54.7	54.5	44.7
Bike	37.4	25.8	31.8	15.4	37.3	26.3	32.7	16.4
Car	73.4	69.5	76.7	48.7	73.3	69.6	76.8	48.8
Motor	41.7	17.0	31.3	19.9	51.4	20.5	37.6	24.0
Person	63.8	46.8	58.0	29.8	59.8	46.5	56.4	28.8
Rider	55.8	34.7	47.0	22.4	54.6	33.1	46.4	20.8
Truck	54.0	50.7	54.3	41.0	54.4	49.8	55.1	40.6

Table 5.13: Class-wise performance analysis of YOLOv8 (ResNet50) with and without **Visual Prompting** under Night-Clear condition

The YOLOv8 with ResNet50 backbone could provide better mAP50 and mAP50-95 for many classes including Bike, Car, Motor, Truck after incororating Visual Prompting. The detection accuracy in the remaining classes dropped a bit due to the use of Visual Prompting. In this scenario, the precision dropped a bit in most classes but the recall value increased for all classes. Specially, the results of Motor class showed a huge improvement across all metrics after adding Visual Prompting.

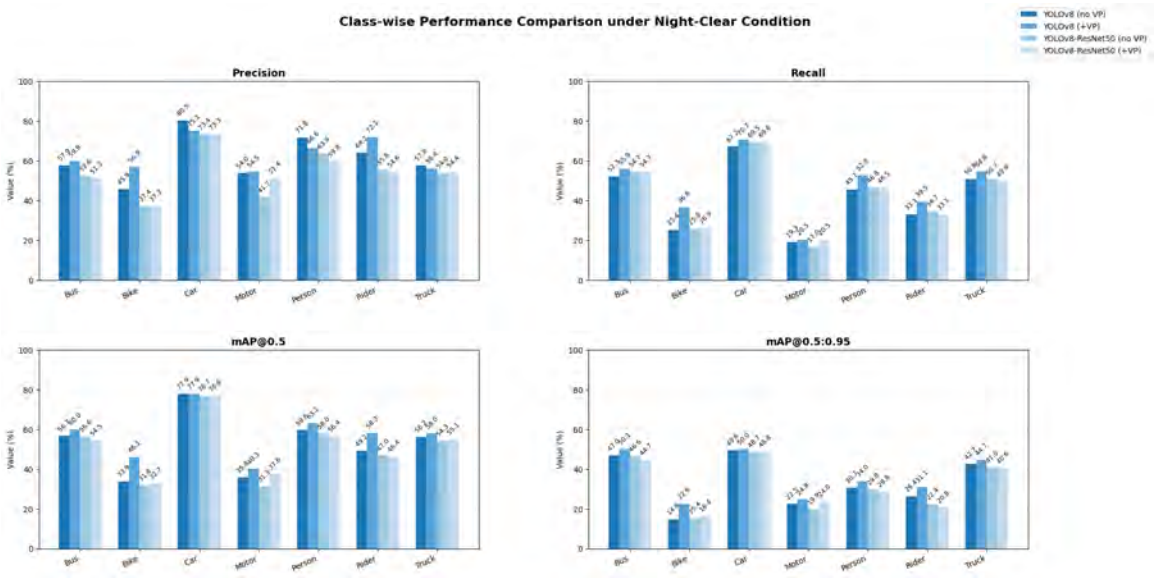


Figure 5.14: Class-wise Performance analysis under **Night-Clear** weather condition

Night-Rainy Condition:

The night-rainy scenario is the most challenging scenario as the models need to deal with both extreme low light and rain. The following table 5.14 shows the

performance of Visual Prompting after integrating with the fine tuning of YOLOv8 model.

Class	YOLOv8 w/o VP				YOLOv8 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	55.0	42.9	44.5	33.9	46.8	40.5	46.3	35.7
Bike	0.0	0.0	1.8	0.5	37.6	12.5	8.7	3.1
Car	76.4	54.0	66.6	38.1	66.3	63.4	69.6	40.8
Motor	58.7	9.1	13.3	6.6	22.3	18.2	19.4	8.2
Person	58.0	13.7	26.1	9.8	43.7	34.7	34.9	13.7
Rider	0.0	0.0	3.4	1.7	14.3	15.4	7.7	4.9
Truck	37.6	34.5	33.7	22.1	44.7	36.4	41.1	26.3

Table 5.14: Class-wise Performance Analysis of **Visual Prompting** Under Night-Rainy condition

Under the difficult Night-Rainy environment, the standard fine tuning of the YOLOv8 model struggled a lot in detecting small objects like Bike and Rider. Visual Prompting demonstrated significant improvements with the baseline YOLOv8 model. The mAP50 and mAP50-95 improved across all classes by integrating Visual Prompting. In this scenario, Visual Prompting demonstrated significant improvement in precision and recall across many classes. However, a few classes faced decline due to incorporating Visual Prompting with YOLOv8.

The class-wise performances of modified YOLOv8 with ResNet50 backbone under night-rainy scenario are presented in the following table 5.15.

Class	YOLOv8+ResNet50 w/o VP				YOLOv8+ResNet50 w/ VP			
	P(%)	R(%)	mAP50(%)	mAP50-95(%)	P(%)	R(%)	mAP50(%)	mAP50-95(%)
Bus	24.4	52.4	36.4	25.4	42.6	42.9	46.3	35.0
Bike	0.0	0.0	0.0	0.0	17.4	6.2	9.9	3.9
Car	61.0	58.7	64.8	36.1	53.7	63.1	66.2	37.8
Motor	80.6	18.2	34.7	17.4	51.4	9.0	24.5	12.3
Person	36.7	19.1	22.4	8.0	37.5	22.0	28.8	9.9
Rider	24.4	7.7	11.1	5.5	33.3	7.6	19.2	9.6
Truck	38.7	31.4	30.5	19.7	28.2	34.5	30.9	21.6

Table 5.15: Class-wise performance analysis of YOLOv8 (ResNet50) with and without **Visual Prompting** under Night-Rainy condition

In this scenario, the modified model has lower performance than the base YOLOv8 model. After adding Visual Prompting, mAP50 and mAP50-95 got improved for most classes. However, the detection accuracy for motor class declined due to integrating Visual Prompting. Even though the precision and recall decreased for some classes, the Visual Prompting technique could still maintain a reasonable precision and recall for all classes.

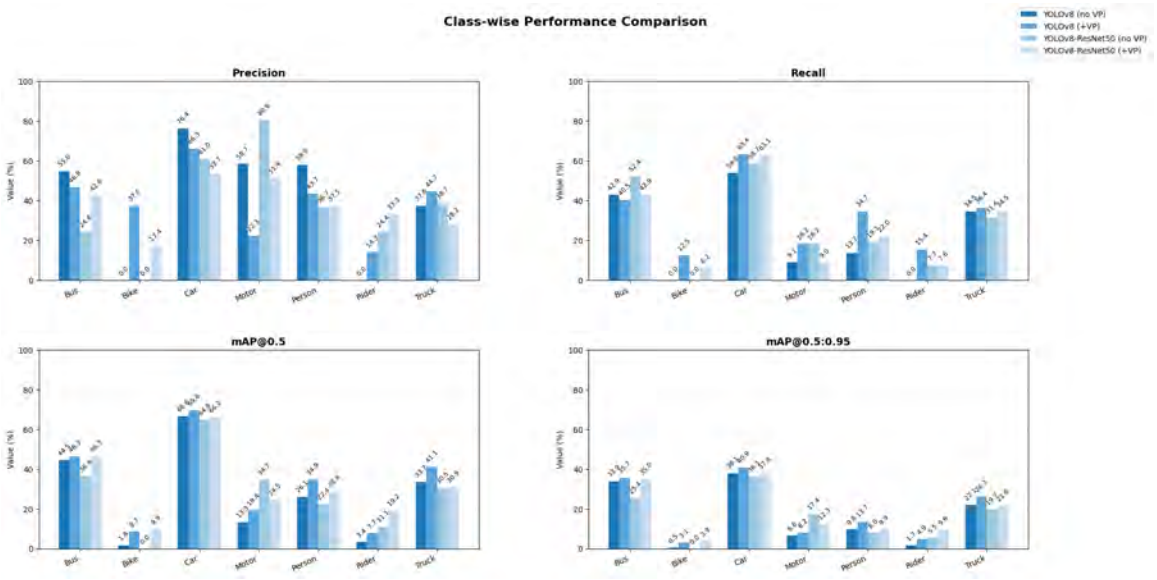


Figure 5.15: Class-wise Performance analysis under **Night-Rainy** weather condition

5.2.4 Discussion

From the results, our proposed method could provide better results for both scenarios. After integrating Visual Prompting, the base YOLOv8 model and the modified YOLOv8 model with ResNet50 backbone demonstrated higher mAP50, mAP50-95 across five weather conditions which shows that the proposed method could improve the detection accuracy of the baseline models. After integrating Visual Prompting, the YOLOv8 model achieved 68.6%, 60.0%, 42.9%, 57.7%, 32.5% mAP50 across daytime-clear, daytime-foggy, dusk-rainy, night-clear and night-rainy conditions respectively. Similarly, the method improved the performance of the modified YOLOv8 with ResNet50 backbone where the mAP50 scores reached to 64.7%, 54.5%,36.6%, 51.3%, 32.2% across the daytime-clear, daytime-foggy, dusk-rainy, night-clear and night-rainy conditions respectively. Moreover, the proposed method could improve the detection accuracy while maintaining a balanced precision and recall. For class-wise performance analysis, the integration of Visual Prompting could benefit most of the classes for both of the models.

Chapter 6

Conclusion

This thesis investigated the effectiveness of visual prompting in order to improve the accuracy of object detection under different weather conditions. While the current methods can show average performance in clear weather, the performance significantly drops under adverse weather like fog, rain and night scenarios. This research addresses the issue by incorporating visual prompting along with the standard fine tuning of existing pretrained models. In order to do so, the YOLOv8 model and the modified YOLOv8 with ResNet50 backbone were utilized and the results showed significant improvement for both models. Moreover, the proposed method adds very little computational overhead compared to existing methods for improving object detection performance across adverse weather. The findings of this research showed visual prompting as a promising technique for improving perception systems of various autonomous systems.

6.1 Summary of Key Findings

The base YOLOv8 model and the modified YOLOv8 model could demonstrate strong detection accuracy in clear weather. In the foggy and night scene, the performance of the models slightly degraded. The performance dropped a lot in complex scenarios like dusk-rainy and night-rainy weather conditions. However, by introducing our visual prompting based method, the performance of the baseline YOLOv8 model and the modified YOLOv8 model got improved across different weather conditions. In general, the YOLOv8 model provided better results than the modified architecture. However, in complex scenarios like night-rainy conditions, the model could show almost similar detection accuracy like the baseline YOLOv8 model while using fewer parameters. For most of the classes across all scenarios, adding visual prompting could enhance the performance of the models while maintaining balanced precision and recall.

6.2 Contributions to the Field

This thesis investigates the adaptation performance of visual prompting mechanisms in Convolutional Neural Network based object detection model YOLOv8. Moreover, the generalizability of this method was also evaluated by integrating this technique with other strong feature extractor. Thus this thesis contributed to the field of object

detection across diverse weather by introducing a new pixel level modification based method. The study provided a new approach for other autonomous systems as it improved the detection accuracy by adding a few parameters. As a result, the approach can be beneficial for the systems that need to deal with diverse weather conditions while having a lightweight design. Moreover, the research also contributed to the field of visual prompting research by extending its implementation to the sector of object detection under diverse weather. Thus the visual prompting based approach can serve as an effective method for enhancing the object detection capabilities of the autonomous systems.

6.3 Limitations

Although the research showed a significant improvement in object detection under various weather conditions, there are some limitations in this research. Due to time constraints, the models could not be trained using different hyperparameter settings and backbone architectures.As a result, the model is not perfectly optimized and the generalizability of visual prompting technique remained untested. Moreover, in real world scenario, the autonomous systems need to deal with more variety of weather conditions. In this research, only five distinct weather conditions have been addressed. Along with that, the models faced difficulty in detecting small and fast moving objects in complex scenarios like night rainy weather. Further, the model could not be deployed in the real-world scenario due to hardware constraints.

6.4 Future Work

In order to overcome our limitations, we have some recommendations for future work. To begin with, we will try with different hyperparameters settings. Moreover, our proposed visual prompting based method will be analyzed with different backbones. Further, we will try to create a mixed synthetic weather balanced dataset in order to solve our data lacking issue. Moreover, we will be able to use a single method across every weather due to the implication of the method.

Bibliography

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, ISSN: 1063-6919, Jun. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074. Accessed: Jan. 30, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6248074/.

[3] S. J. Jia, Y. T. Zhai, and X. W. Jia, "Detection of traffic and road condition based on svm and phow," in *Applied Science, Materials Science and Information Technologies in Industry*, ser. Applied Mechanics and Materials, vol. 513, Trans Tech Publications Ltd, May 2014, pp. 3651–3654. DOI: 10.4028/www.scientific.net/AMM.513-517.3651.

[4] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[5] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015. DOI: 10.1038/nature14542. [Online]. Available: https://www.nature.com/articles/nature14542.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015. DOI: 10.48550/arXiv.1512.03385. [Online]. Available: https://arxiv.org/abs/1512.03385.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016. arXiv: 1506.02640 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1506.02640.

[8] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic Foggy Scene Understanding with Synthetic Data," en, *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, Sep. 2018, ISSN: 1573-1405. DOI: 10.1007/s11263-018-1072-8. Accessed: Jan. 30, 2025. [Online]. Available: https://doi.org/10.1007/s11263-018-1072-8.

[9] F. Yu et al., "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," 2020, pp. 2636–2645. Accessed: Jan. 30, 2025. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Yu_BDD100K_A_Diverse_Driving_Dataset_for_Heterogeneous_Multitask_Learning_CVPR_2020_paper.html.

[10] A. Radford et al., "Learning transferable visual models from natural language supervision," *CoRR*, 2021. arXiv: 2103.00020.

[11] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola. "Exploring visual prompts for adapting large-scale models." arXiv: 2203.17274.

[12] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, *Exploring visual prompts for adapting large-scale models*, https://arXiv.org/abs/2203.17274, arXiv preprint, arXiv:2203.17274 [cs.CV], 2022. DOI: 10.48550/arXiv.2203.17274.

[13] R.-A. Bratulescu, R.-I. Vatasoiu, G. Sucic, S.-A. Mitroi, M.-C. Vochin, and M.-A. Sachian, "Object detection in autonomous vehicles," in *2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2022, pp. 375–380. DOI: 10.1109/WPMC55625.2022.10014804.

[14] O. Hmidani and E. M. Ismaili Alaoui, "A comprehensive survey of the r-cnn family for object detection," in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2022, pp. 1–6. DOI: 10.1109/CommNet56067.2022.9993862.

[15] M. Jia et al., "Visual prompt tuning," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 709–727, ISBN: 978-3-031-19827-4.

[16] Y. Rao et al., *Denseclip: Language-guided dense prediction with context-aware prompting*, arXiv:2112.01518 [cs], Mar. 2022. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2112.01518.

[17] A. Chen, Y. Yao, P.-Y. Chen, Y. Zhang, and S. Liu, "Understanding and improving visual prompting: A label-mapping perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, arXiv preprint arXiv:2211.11635, 2023, pp. 1234–1243. DOI: 10.48550/arXiv.2211.11635. [Online]. Available: https://arxiv.org/abs/2211.11635.

[18] Y. Jiang et al., *Vima: General robot manipulation with multimodal prompts*, arXiv:2210.03094 [cs], May 2023. DOI: 10.48550/arXiv.2210.03094. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2210.03094.

[19] G. Jocher et al., *Yolov8: The latest version of the yolo object detector*, https://docs.ultralytics.com/, Ultralytics Official Documentation, 2023.

[20] A. Kirillov et al., *Segment anything*, arXiv:2304.02643 [cs], Apr. 2023. arXiv: arXiv:2304.02643. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2304.02643.

[21] Labellerr, *Vrp-sam: Advanced image segmentation with visual reference prompts*, 2023. [Online]. Available: https://www.labellerr.com/blog/vrp-sam-advanced-image-segmentation/#architecture-of-vrp-sam.

[22] F. Li et al., "Visual in-context prompting," *CoRR*, vol. abs/2311.13601, Nov. 2023, arXiv preprint. DOI: 10.48550/arXiv.2311.13601. arXiv: 2311.13601. [Online]. Available: https://arxiv.org/abs/2311.13601.

[23] K. Li, P. Wang, X. Li, and L. Cui, "Rethinking and improving visual prompt selection for in-context learning segmentation," *arXiv preprint*, 2023. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2303.08138.

[24] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, Jan. 2023, ISSN: 0360-0300. DOI: 10.1145/3560815. [Online]. Available: https://doi.org/10.1145/3560815.

[25] T. Liu, Y. Hu, W. Wu, Y. Wang, K. Xu, and Q. Yin, *Prompt-based context- and domain-aware pretraining for vision and language navigation*, arXiv:2309.03661 [cs], Dec. 2023. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2309.03661.

[26] R. Rajendran, T. Nguyen, and W. Chen, "Visual prompting for generalized few-shot segmentation: A multi-scale approach," *Papers with Code*, 2023. [Online]. Available: https://paperswithcode.com/paper/visual-prompting-for-generalized-few-shot.

[27] K. Shankar, S. Baskar, et al., "Enhanced object detection using advanced yolo models in complex environments," *Engineering Letters*, vol. 33, no. 6, pp. xx–xx, 2023, Accessed: 2025-06-16. [Online]. Available: https://www.engineeringletters.com/issues_v33/issue_6/EL_33_6_09.pdf.

[28] H.-A. Tsao, L. Hsiung, P.-Y. Chen, S. Liu, and T.-Y. Ho, *Autovp: An automated visual prompting framework and benchmark*, arXiv preprint arXiv:2310.08381, 2023. arXiv: 2310.08381 [cs.CV]. [Online]. Available: https://doi.org/10.48550/arXiv.2310.08381.

[29] H. Wang, T. Liu, J. Chen, C. Fan, Y. Qin, and J. Han, "Full-Perception Robotic Surgery Environment with Anti-Occlusion Global–Local Joint Positioning," en, *Sensors*, vol. 23, no. 20, p. 8637, Jan. 2023, ISSN: 1424-8220. DOI: 10.3390/s23208637. Accessed: Oct. 18, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/23/20/8637.

[30] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, *SegGPT: Segmenting Everything In Context*, arXiv:2304.03284 [cs], Apr. 2023. DOI: 10.48550/arXiv.2304.03284. arXiv: arXiv:2304.03284. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2304.03284.

[31] Y. Wang, X. Zhang, and L. Liu, "Vlm-grounder: A vlm agent for zero-shot 3d visual grounding," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. [Online]. Available: https://openreview.net/pdf?id=IcOrwlXzMi.

[32] L. Yang, Y. Wang, X. Li, X. Wang, and J. Yang, *Fine-grained visual prompting*, arXiv:2306.04356 [cs], Dec. 2023. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2306.04356.

[33] S.-F. Chen, J.-C. Chen, I.-H. Jhuo, and Y.-Y. Lin, *Improving Visual Object Tracking through Visual Prompting*, arXiv:2409.18901 [cs, eess], Sep. 2024. DOI: 10.48550/arXiv.2409.18901. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2409.18901.

[34] J. Doe, A. Smith, and K. Lee, "Real-time object detection with enhanced rt-detr architecture," in *Proceedings of the IEEE Conference*, Accessed: 2025-06-16, IEEE, 2024, pp. 1–6. DOI: 10.1109/10756955. [Online]. Available: https://ieeexplore.ieee.org/document/10756955.

[35] H. Gupta, O. Kotlyar, H. Andreasson, and A. J. Lilienthal, "Robust object detection in challenging weather conditions," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 7508–7517. DOI: 10.1109/WACV57701.2024.00735.

[36] M. Jeon, G.-M. Park, and H. Hwang, "Fisheye Object Detection with Visual Prompting-Aided Fine-Tuning," en, *Remote Sensing*, vol. 16, no. 12, p. 2054, Jun. 2024, ISSN: 2072-4292. DOI: 10.3390/rs16122054. Accessed: Oct. 17, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/12/2054.

[37] Q. Jiang, F. Li, Z. Zeng, T. Ren, S. Liu, and L. Zhang, *T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy*, arXiv:2403.14610 [cs], Mar. 2024. DOI: 10.48550/arXiv.2403.14610. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2403.14610.

[38] J. Kim, H. Park, and S. Lee, "Vlm-pl: Advanced pseudo labeling approach for class incremental object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024. [Online]. Available: https://arxiv.org/abs/2403.05346.

[39] X. Li, Y. Zhang, and L. Wang, "Rt-detr++: Improved real-time detection transformer for object detection," *arXiv preprint*, vol. arXiv:2412.13490, 2024, Accessed: 2025-06-16. [Online]. Available: https://arxiv.org/pdf/2412.13490v1.

[40] F. Liu, K. Fang, P. Abbeel, and S. Levine, *MOKA: Open-World Robotic Manipulation through Mark-Based Visual Prompting*, arXiv:2403.03174 [cs], Sep. 2024. DOI: 10.48550/arXiv.2403.03174. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2403.03174.

[41] F. Neha, D. Bhati, D. K. Shukla, and M. Amiruzzaman, "From classical techniques to convolution-based models: A review of object detection algorithms," *arXiv preprint*, Dec. 2024. arXiv: 2412.05252 [cs.CV].

[42] R. Rezaei, M. J. Sabet, J. Gu, D. Rueckert, P. Torr, and A. Khakzar. "Learning visual prompts for guiding the attention of vision transformers." arXiv: 2406.03303.

[43] N. U. A. Tahir, Z. Zhang, M. Asim, J. Chen, and M. ELAffendi, "Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches," *Algorithms*, vol. 17, no. 3, 2024, ISSN: 1999-4893. DOI: 10.3390/a17030103. [Online]. Available: https://www.mdpi.com/1999-4893/17/3/103.

[44] M. Yaseen, "What is yolov8: An in-depth exploration of the internal features of the next-generation object detector," *arXiv preprint arXiv:2408.15857*, 2024.

[45] Y. Zhang, C. Li, L. Zhang, and J. He, "Advanced pseudo labeling approach for class incremental object detection via vision-language model," *arXiv preprint*, 2024. [Online]. Available: https://www.arxiv.org/abs/2407.10233.

[46] H. Zhao et al., *Real-time object detection and robotic manipulation for agriculture using a YOLO-based learning approach*, arXiv:2401.15785 [cs], Jan. 2024. DOI: 10.48550/arXiv.2401.15785. Accessed: Oct. 18, 2024. [Online]. Available: http://arxiv.org/abs/2401.15785.

[47] C. Jin et al., "Lor-vp: Low-rank visual prompting for efficient vision model adaptation," in *Proc. of the International Conference on Representation Learning (ICLR 2025)*, arXiv:2502.00896 [cs.CV], 2025. DOI: 10.48550/arXiv.2502.00896. [Online]. Available: https://arxiv.org/abs/2502.00896.

[48] B. Liu, J. Jin, Y. Zhang, and C. Sun, "WRRT-DETR: Weather-Robust RT-DETR for Drone-View Object Detection in Adverse Weather," *Drones*, vol. 9, no. 5, p. 369, 2025, These authors contributed equally to this work. DOI: 10.3390/drones9050369. [Online]. Available: https://doi.org/10.3390/drones9050369.

[49] Z. Wang et al., *A comprehensive survey on data augmentation*, 2025. arXiv: 2405.09591 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2405.09591.