Object Detection for Autonomous Vehicles in Adverse Weather and Poor Lighting

CS402 - Major Project Report

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering

by

Gollapalli Harshitha (221CS124)

Hitha N (221CS130)

under the guidance of

Dr. Basavaraj Talawar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE - 575025
November, 2025

DECLARATION

I hereby declare that the **CS402 - Major Project** report entitled **Object Detection for Autonomous Vehicles in Adverse Weather and Poor Lighting** being submitted to the Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, in fulfilment of the requirements of the B.Tech in CSE is a bonafide report of the work carried out by us. The material contained in this Report has not been submitted to any University or Institution for the award of any degree.

Gollapalli Harshitha (221CS124)
Hitha N (211CS130)
Department of Computer Science and Engineering
NITK, Surathkal

Place: NITK, Surathkal.
Date: 28-11-2025

CERTIFICATE

This is to certify that the **CS402 - Major Project** report entitled **Object Detection for Autonomous Vehicles in Adverse Weather and Poor Lighting** submitted by **Gollapalli Harshitha (221CS124)** and **Hitha N (221CS130)**, as the record of the work carried out by the team is accepted as the CS402 - Major Project work report submission in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering, NITK, Surathkal.

Dr. Basavaraj Talawar
Department of Computer Science and Engineering,
NITK, Surathkal

Abstract

Autonomous vehicles rely on camera-based perception systems to detect surrounding objects and navigate safely. However, their performance degrades significantly under adverse weather conditions such as fog, rain, and snow, as well as in low-illumination settings including nighttime and shadowed environments. This project develops a robust object-detection framework designed to sustain high accuracy in such challenging scenarios. Using the IDD-AW dataset, which contains diverse real-world driving scenes captured in harsh weather and poor lighting, we perform dataset preprocessing, weather-oriented augmentations, and transfer learning to enhance model adaptability. The YOLOv8 architecture is fine-tuned to learn weather-specific visual patterns, enabling improved detection stability under low contrast and reduced visibility. Experimental results demonstrate substantial performance gains over baseline models, highlighting the effectiveness of weather-focused training strategies. This system contributes to safer autonomous navigation by offering consistent and reliable object detection across adverse environmental conditions.

Keywords: Autonomous Vehicles, Object Detection, Adverse Weather, Low-Light Conditions, IDD-AW Dataset, YOLOv8, Transfer Learning, Robust Perception Systems

Contents

List of Tables

List of Figures

1 Introduction

Autonomous vehicles depend heavily on perception systems to understand their surroundings and operate safely. Among all sensing approaches, camera-based vision is one of the most essential because it captures rich visual cues necessary for detecting vehicles, pedestrians, road signs, and other critical objects. However, real-world driving environments are often affected by adverse weather such as rain, fog, and snow, as well as low-light conditions including nighttime or deep shadows. These challenging environmental factors degrade visibility and significantly reduce the performance of conventional object detection systems. This project focuses on developing a robust object detection framework that can maintain strong performance even under visually degraded conditions. Leveraging the IDD-AW dataset, which contains diverse adverse-weather scenes from Indian roads, the system aims to improve detection reliability for autonomous navigation in challenging environments.

1.1 Core Concepts of Object Detection

Object detection is the computer vision task of identifying and localizing objects within an image. Formally, an image can be represented as a function I(x, y) defining pixel intensities, while the detection output consists of bounding boxes and class labels for objects of interest. Object detection systems rely on several foundational ideas:

- **Bounding Boxes**: Each detected object is represented by a rectangular box defined by coordinates (x, y, w, h), where the aim is to tightly enclose the target.
- **Feature Extraction**: Deep neural networks extract hierarchical features from edges to high-level shapes to recognize object categories even in complex scenes.
- **Confidence Scores**: Each prediction includes a probability score indicating the likelihood of an object's presence.
- **Localization and Classification**: A model must correctly *identify* an object's class and *localize* it accurately.

Modern deep-learning models like YOLO, SSD, and Faster R-CNN have achieved remarkable performance in clear-weather conditions, establishing them as key components for autonomous driving perception.

1.2 Challenges in Adverse Weather and Low Lighting

Adverse weather and poor lighting conditions introduce distortions that significantly hinder visual perception. These conditions create several major obstacles for object detection systems:

- **Rain:** Raindrops cause streaks and motion blur, reducing contrast and obscuring object edges.
- **Fog and Haze:** Scattering of light reduces visibility, introduces washout effects, and compresses depth cues.
- **Snow:** Snowfall introduces occlusion, noise, and random bright spots across the scene.
- **Nighttime and Low Light:** High noise levels, low contrast, shadows, and overexposure from headlights impact detection performance.

A model trained primarily on clear-weather datasets encounters a *domain shift* when exposed to degraded visual conditions, causing misclassifications and missed detections. Ensuring robust performance under these challenges is essential for safe autonomous driving.

1.3 Object Detection Model Architecture

Contemporary object detection architectures can be broadly categorized into two families:

- **One-Stage Detectors:** Models such as YOLO and SSD perform detection in a single step, offering real-time speed—an essential requirement for autonomous vehicles. They predict bounding boxes and class scores directly from feature maps.

- **Two-Stage Detectors:** Models such as Faster R-CNN first generate region proposals They offer high accuracy but comparatively lower inference speed.

- **YOLOv8 Framework:**

  YOLOv8, used in this project, offers several advantages:

- Anchor-free detection
- Feature pyramid structures for multi-scale detection
- Advanced augmentation techniques
- Strong performance in both accuracy and speed

These characteristics make it suitable for handling noisy or degraded images from adverse environments.

1.4 Historical Insights and Importance of Weather Robust Detection

The evolution of object detection began with handcrafted features such as Haar cascades and HOG+SVM, progressing to deep learning–based detectors like R-CNN, YOLO, and Retina Net. While early models performed well in controlled conditions, their limitations in real-world environments highlighted the need for more adaptable detection strategies.

Research on weather-robust perception gained momentum as autonomous driving technologies matured. Adverse-weather datasets such as Foggy Cityscapes, Rainy Driving, and IDD-AW were introduced to study how degraded visibility impacts detection algorithms. These datasets helped drive advancements in:

- Weather-aware augmentation
- Domain adaptation techniques
- Multi-modal perception approaches

As autonomous vehicles move closer to widespread adoption, resilient object detection under challenging weather and lighting conditions is becoming increasingly critical for safety and reliability.

1.5 Current Research Directions

Current research in weather-resilient object detection focuses on several promising directions:

- **Domain Adaptation:** Bridging the gap between clear-weather training data and adverse-weather test data.
- **Image Restoration + Detection Pipelines:** Using dehazing, deraining, or low-light enhancement before detection.
- **Weather-Condition-Specific Models:** Training separate detection models tailored to different weather types.
- **End-to-End Adverse-Weather Training:** Integrating weather effects directly in the augmentation pipeline.
- **Transformer-Based Detectors:** Leveraging architectures like DETR for better robustness to noise and distortions.
- **Synthetic Weather Generation:** Using generative models (e.g., GANs) to create realistic weather conditions for training.

These developments aim to create systems capable of reliable perception even in the harsh environmental scenarios autonomous vehicles frequently encounter.

2 Literature Review

2.1 Overview of Object Detection in Autonomous Vehicles

Object detection is a fundamental component of autonomous driving, enabling the identification and localization of critical road entities such as vehicles, pedestrians, traffic signs, cyclists, animals, and static obstacles. Modern autonomous systems rely extensively on deep learning–based computer vision models that process camera inputs and output bounding boxes with corresponding class labels.

Traditional object detection techniques, including Viola Jones, Haar Cascades, and HOG SVM, served as early methods but suffered from limited robustness and poor adaptability to environmental variations. The emergence of deep learning revolutionized object detection through models such as R-CNN, Fast R-CNN, Faster R-CNN, SSD, Retina-Net, and the YOLO family. Among these, YOLO gained prominence due to its unified architecture and real-time performance, making it highly suitable for autonomous driving applications.

Despite the advancements, most existing models are trained on clear-weather datasets, resulting in significant performance degradation under adverse weather or poor lighting conditions.

2.2 Impact of Adverse Weather on Vision-Based Perception

Adverse weather significantly affects camera visibility and image quality. Weather-induced distortions include:

Fog and Smog

- Light scattering
- Low contrast and washed-out visuals
- Reduced object boundary clarity

Rain

- Raindrop streaks and lens distortions
- Motion blur
- Reflections from wet surfaces

Snow

- Visual occlusions caused by falling snow
- Reduced visibility
- Strong reflections and scene brightness changes

Low-Light and Nighttime Conditions

- Increased noise and grain
- Underexposure
- Loss of colour information
- Glare from headlights or street lights

These degradations introduce discrepancies between training data (typically clear-weather) and real-world conditions, causing models to produce false detections, missed objects, and overall reduced reliability.

2.3 Challenges in Current Object Detection Systems

Object detection models exhibit several limitations when deployed in adverse weather and poor lighting:

1. **Limited Weather Diversity in Training Data**
   Popular datasets such as COCO, KITTI, and Cityscapes contain insufficient fog, rain, snow, or nighttime imagery.

2. **Domain Shift**

   Performance drops significantly when the model encounters visual domains not seen during training.

3. **Reduced Generalization**

   Deep models overfit to clear-weather features and fail to interpret degraded patterns.

4. **High Misclassification Rates**

   Artifacts like rain streaks, glare, and shadows often resemble object-like shapes, confusing detectors.

5. **Insufficient Annotated Data in Harsh Weather**

   Collecting and annotating data under severe conditions is challenging and time-consuming.

6. **Unstructured Traffic Conditions** Regions like India exhibit non-lane-disciplined behavior, mixed traffic patterns, and irregular obstacles, further complicating detection.

These challenges highlight the need for weather-resilient datasets and robust detection frameworks.

2.4 Review of Existing Adverse-Weather Datasets

To address weather-related vision challenges, several datasets have been proposed:

- **ACDC:** Contains real-world rain, fog, snow, and nighttime scenes from European cities.
- **DAWN:** Focuses on sandstorms and rain but has a limited number of samples.
- **Foggy Cityscapes / Rain Cityscapes:** Synthetic fog and rain overlays applied to the Cityscapes dataset.
- **BDD100K:** Offers weather annotations but lacks substantial representation of extreme conditions.

While these datasets contribute valuable insights, they do not fully capture the complexity of unstructured traffic or severe real-world weather scenarios.

2.5 The IDD-AW Dataset and Its Significance

The IDD-AW (Indian Driving Dataset – Adverse Weather) is designed to overcome the limitations of existing datasets. It provides extensive coverage of diverse Indian driving conditions, including:

- Real fog
- Heavy rain
- Snow-covered scenes
- Low-light and nighttime environments
- Dense, unstructured traffic
- Complex road geometries

The dataset presents several unique challenges:

- Objects appear partially visible or heavily occluded.
- Traffic includes a wide variety of classes not present in typical Western datasets.
- Scenes contain cluttered and irregular backgrounds.
- Lighting conditions vary abruptly.
- Pedestrian and vehicle movement is unpredictable.

IDD-AW's diversity and complexity make it ideal for developing robust object detection models for harsh weather and unstructured traffic environments.

2.6 Evolution of Deep Learning–Based Object Detectors

Deep learning methods have significantly advanced object detection:

Two-Stage Detectors

- **R-CNN, Fast R-CNN, Faster R-CNN**
  High accuracy but computationally slow; unsuitable for real-time autonomous driving.

One-Stage Detectors

- **YOLO, SSD, Retina-Net**
  Offer real-time performance and improved efficiency.
  Continuous improvements in the YOLO family (v3 to v8) provide:
    - Enhanced feature extraction

- Multi-scale detection
- Higher accuracy
- Lower inference latency
- More lightweight architectures

YOLOv8, used in this project, integrates advanced convolutional blocks and optimized training strategies, making it more resilient to poor lighting and complex visual conditions.

2.7 Weather-Specific Image Enhancement Techniques

Parallel to detection models, research also explores image restoration approaches such as:

1. **Dehazing** – Removes fog using atmospheric scattering models.
2. **Deraining** – Reduces rain streaks using CNN or GAN-based methods.
3. **Desnowing** – Uses multi-scale filters to restore snow-affected images.
4. **Low-Light Enhancement** – Improves illumination and reduces noise in dark scenes.

Although effective in improving visibility, these methods often introduce artifacts and impose computational overhead, making them less suitable for real-time autonomous driving. Training detectors directly on adverse-weather data is generally more efficient.

2.8 Research Gaps

Despite progress, key gaps remain:

- Limited availability of real-world adverse-weather datasets
- Underrepresentation of unstructured traffic scenarios
- Lack of large-scale examples of fog, snow, and nighttime scenes
- Models not specifically optimized for harsh environments
- Few studies combining diverse weather and complex traffic patterns
- Limited exploration of safety-critical performance metrics

This project aims to address these gaps by leveraging the IDD-AW dataset and developing a custom YOLOv8-based object detection system optimized for challenging conditions.

2.9 Summary

This chapter reviewed key advancements in object detection, challenges posed by adverse weather and poor lighting, and recent datasets addressing atmospheric degradation. The literature indicates that traditional and even modern detectors struggle when visibility deteriorates due to insufficient training on adverse-weather data. IDD-AW provides a comprehensive solution by offering extensive, real-world, weather-degraded imagery in unstructured traffic environments.

3 SYSTEM DESIGN AND ARCHITECTURE

3.1 System Overview

The proposed system is designed to perform robust and reliable object detection for autonomous vehicles operating under adverse weather conditions and poor lighting environments. To achieve this, the architecture integrates dataset processing, preprocessing, model development, prediction, and evaluation into a unified and well structured workflow.

The system operates through three primary stages:

1. Dataset Processing and Preparation

- Extracting fog, rain, snow, and low-light images from the IDD-AW dataset
- Organizing the dataset and standardizing annotation formats
- Applying weather-relevant augmentations to enhance model robustness

2. Model Development Using YOLOv8

- Utilizing pre-trained YOLOv8 weights
- Applying transfer learning for domain adaptation
- Fine-tuning the model for adverse-weather perception

3. Evaluation and Deployment

- Testing detection performance on challenging weather conditions
- Generating detection outputs and predictions
- Assessing system robustness under multiple environmental variations

The overall goal is to produce an object detection system optimized for degraded visibility scenarios, improving the safety and reliability of autonomous driving.

3.2 Design Goals and Considerations

The system is developed with the following design objectives:

1. Weather Robustness

The model must maintain high detection accuracy in fog, rain, snow, and low-light conditions, minimizing the impact of noise, occlusions, and low visibility.

2. High Accuracy

Using advanced deep learning techniques and fine-tuned YOLOv8 architecture, the model targets high mAP scores across all conditions.

3. Real-Time Efficiency

Autonomous vehicles require real-time perception. YOLOv8, with its optimized inference speed, ensures suitability for on-road deployment.

4. Scalability

The system supports future expansion, such as:

- Additional object classes
- Larger datasets
- Integration with sensors such as LiDAR and RADAR

5. Compatibility

The designed architecture seamlessly integrates with autonomous driving modules such as:

- Object tracking
- Motion prediction
- Path planning
- Vehicle control

3.3 System Architecture

The system consists of four major layers that work cohesively throughout the workflow:

A. Input Layer – Dataset Integration

- Import the IDD-AW (19 GB) dataset
- Extract weather-specific frames (fog, rain, snow, low-light)
- Convert annotations to YOLO format
- Split data into training, validation, and testing subsets

B. Preprocessing and Augmentation Layer

- Resize images to suitable input dimensions (e.g., 640×640 or 1280×1280)
- Apply augmentation techniques such as:
    - Contrast reduction
    - Noise injection
    - Fog simulation
    - Brightness adjustments
    - Motion blur
- Ensure sufficient variation across weather and lighting scenarios

C. YOLOv8-Based Detection Model Layer

- Load pre-trained YOLOv8 backbone
- Fine-tune using IDD-AW images
- Optimize hyperparameters such as:
    - Batch size
    - Learning rate
    - Confidence thresholds
- Train model on GPU-enabled platforms

D. Output Layer

- Generate bounding box predictions
- Assign class labels and confidence values
- Visualize detection results
- Export evaluation metrics (mAP, precision, recall)

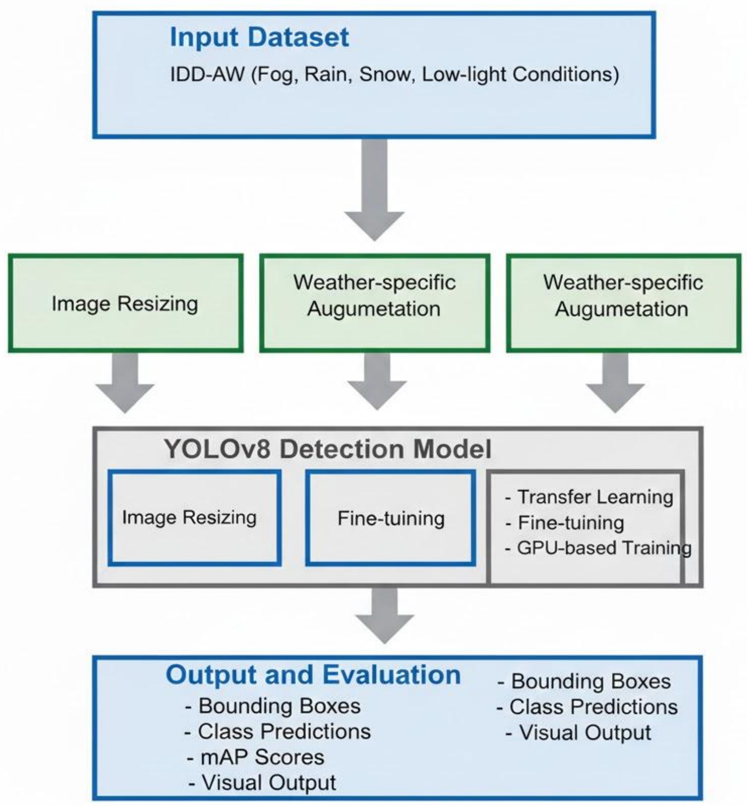

Fig.1 Architecture Summary Diagram

3.4 System Components and Modules

1. Dataset Module

Handles all dataset-related operations:

- Loading and organizing IDD-AW images
- Reading annotation files
- Converting labels to YOLO text format
- Creating train/val/test splits

2. Preprocessing Module

Enhances image quality and improves model learning:

- Image resizing
- Normalization
- Weather-based augmentations such as:
  - Fog simulation
  - Snow overlays
  - Brightness reduction
  - Motion blur
  - Noise addition

3. Model Training Module

Implements the YOLOv8 training pipeline:

- Loading pre-trained weights
- Training for multiple epochs
- Monitoring training curves
- Saving best-performing checkpoints

4. Detection Module

Runs inference on new or test images:

- Generating bounding boxes
- Estimating class confidence
- Filtering low-confidence predictions

5. Evaluation Module

Assesses the model's performance using:

- mAP50
- mAP50–95
- Precision
- Recall
- Weather-wise accuracy breakdown

3.5 Data Flow Diagram (DFD)

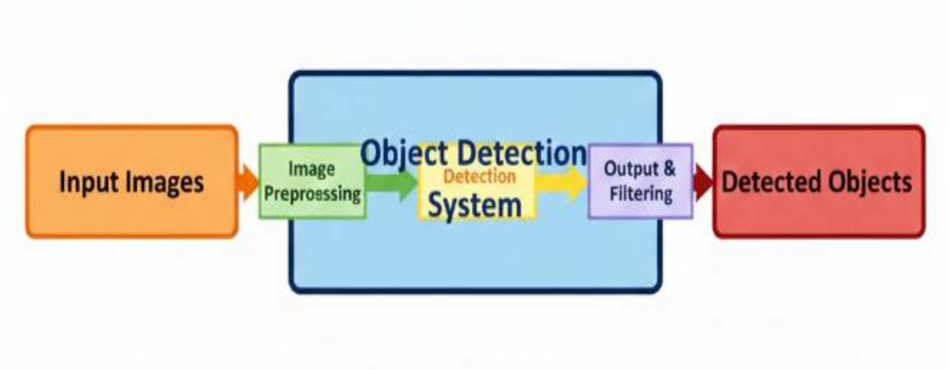

Fig.2 Level 0 DFD

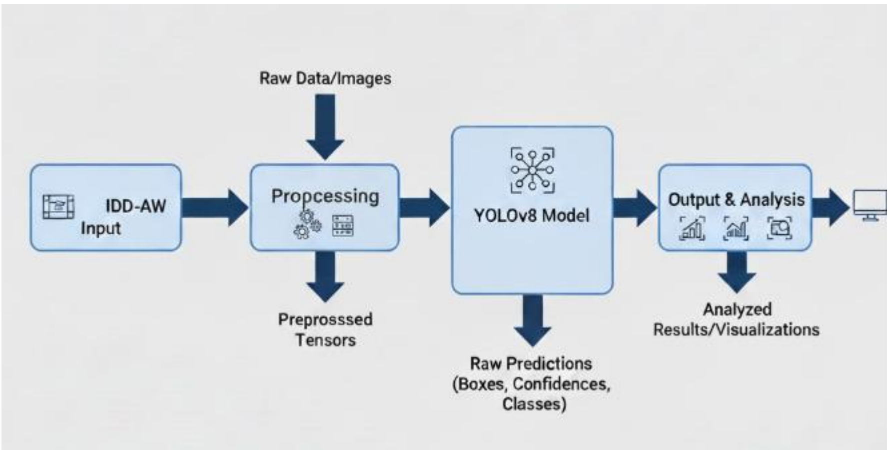

Fig.3 Level 1 DFD

3.6 Use Case Diagram

Actors

- Autonomous Vehicle System (Primary User)
- Dataset Engineer
- Model Trainer

Use Cases

- Load and preprocess dataset
- Train object detection model
- Test model under adverse weather
- Evaluate detection accuracy

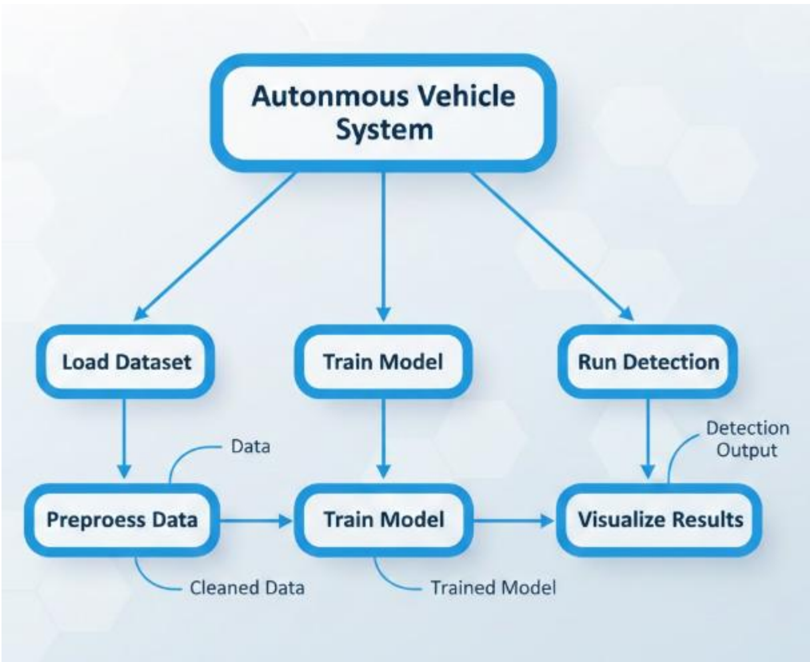

Fig.4 Use case Diagram

3.7 Hardware and Software Requirements

Hardware Requirements

- GPU-enabled workstation or cloud platform
- NVIDIA GPU (8–16 GB VRAM recommended)
- 16–32 GB system RAM
- Minimum 100 GB free storage (dataset + models)

Recommended platforms:

- Google Colab Pro / Pro+
- Kaggle Notebooks
- Local machine with CUDA support

Software Requirements

- OS: Windows / Linux / Ubuntu
- Programming Language: Python 3.8+
- Deep Learning Framework: PyTorch (YOLO backend)
-

Libraries and Tools:

- Ultralytics YOLOv8
- NumPy
- OpenCV
- Pandas
- Matplotlib
- tqdm
- PyTorch CUDA Toolkit

4 METHODOLOGY

4.1 Proposed Approach

The primary objective of this research is to design a **weather-resilient object detection system** capable of performing reliably under fog, rain, snow, and low-light conditions. To accomplish this, the methodology integrates a specialized adverse-weather dataset with a state-of-the-art deep learning architecture. The overall workflow consists of the following stages:

1. **Dataset Extraction and Preparation**
   Utilizing the **IDD-AW dataset (19 GB)** containing real-world Indian driving scenes captured under various adverse weather and illumination conditions.
2. **Annotation Formatting**
   Converting the dataset's annotation format into YOLO-compatible text files.
3. **Data Preprocessing and Augmentation**
   Applying weather-focused transformations to improve model generalization.
4. **Model Selection and Training**
   Training YOLOv8 through transfer learning and fine-tuning on the IDD-AW dataset.
5. **Testing and Evaluation**
   Evaluating detection accuracy using mAP metrics, precision–recall scores, and condition-wise performance breakdown.

This methodological pipeline ensures that the detector learns robust visual representations even in low-visibility conditions, thereby improving performance in actual autonomous driving scenarios.

4.2 Algorithm Design and Description

The algorithm powering the system is **YOLOv8**, one of the leading one-stage object detectors known for its efficiency, accuracy, and real-time inference capabilities. The following subsections describe its structure and relevance to the proposed design.

4.2.1 YOLOv8 Architecture Overview

YOLOv8 consists of three core components:

1. Backbone

- Extracts low-level and high-level visual features.
- Uses optimized convolutional blocks for high speed and accuracy.
- Learns robust representations suitable for weather-degraded images.

2. Neck

- Combines multi-scale features using a Feature Pyramid Network (FPN).
- Ensures improved detection of objects with varying sizes (e.g., pedestrians, distant vehicles).

3. Head

- Produces bounding boxes, class labels, and confidence scores.
- Utilizes a single forward pass for real-time prediction.

4.2.2 Rationale for Selecting YOLOv8

YOLOv8 is chosen due to the following advantages:

- Strong performance on **low-resolution and noisy inputs**
- High-speed inference suitable for real-time autonomous systems
- Robust feature extraction under fog, rain, snow, and low-light distortions
- Support for transfer learning and custom dataset training
- Efficient optimization and simplified deployment pipeline

4.2.3 YOLOv8 Workflow

The YOLOv8 inference workflow follows:

1. Load input image
2. Resize to network input size
3. Extract features using backbone
4. Combine features in neck
5. Predict objects using head
6. Apply Non-Maximum Suppression (NMS)
7. Output bounding boxes, classes, and confidence scores

This streamlined pipeline makes YOLOv8 highly suitable for challenging weather conditions.

4.3 Implementation Steps

The implementation is performed through the following sequential steps:

Step 1: Dataset Preparation

- Download and extract the **IDD-AW (19 GB)** dataset.
- Organize images based on weather categories: fog, rain, snow, and low-light.
- Inspect annotation structure and image quality.
- Split dataset into training, validation, and test sets (e.g., 70% / 20% / 10%).

Step 2: Data Preprocessing

- Resize images to **640×640** (or **1280×1280** if required).
- Normalize pixel intensity values.
- Remove corrupted or incomplete samples.
- Maintain standardized directory structure:

```
dataset/
    images/train/
    images/valid/
    images/test/
    labels/train/
    labels/valid/
    labels/test/
```

Step 3: Data Augmentation

Weather-focused augmentations are applied to increase diversity and robustness:

Intensity Modifications
- Brightness reduction
- Exposure variation
- Contrast suppression

Blur-Based Augmentations
- Gaussian blur
- Motion blur (rain-like streak effects)

Noise and Distortions
- Fog simulation
- Snow particle overlays
- Color desaturation
- Random shadow simulation

These augmentations emulate real-world distortions, improving model adaptability to unseen scenarios.

Step 4: Model Selection

- Choose YOLOv8 variant (YOLOv8n, YOLOv8s, YOLOv8m) based on available GPU resources.
- Load pre-trained weights for transfer learning.
- Configure key hyperparameters:
    - Learning rate
    - Batch size
    - Number of epochs
    - Image size
    - Confidence threshold

Step 5: Model Training

Training is performed using GPU acceleration. The model training loop includes:

- Forward propagation
- Loss computation
- Backpropagation
- Weight updates

Monitoring metrics:

- Classification loss
- Localization loss
- Validation mAP

Early stopping is applied when model improvement stabilizes. Best-performing weights are automatically saved.

Step 6: Model Testing and Evaluation

After training:

- Run inference on test set images
- Visualize predicted bounding boxes
- Compute evaluation metrics:
    - **mAP50**
    - **mAP50–95**
    - **Precision–Recall curves**

Weather-wise accuracy comparison is performed to analyze robustness in fog, rain, snow, and low-light.

Step 7: Optimization and Fine-Tuning

If accuracy is insufficient:

- Tune learning rate
- Increase epochs
- Strengthen augmentations
- Switch to larger YOLOv8 models
- Balance dataset categories

Iterative fine-tuning improves both accuracy and stability.

4.4 Flowchart of the Proposed System

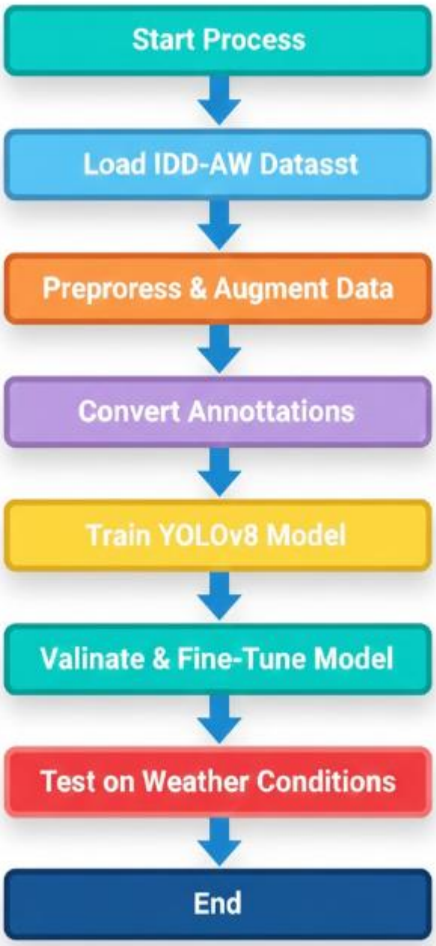

Fig.5

4.5 Weather Robustness Features Implemented

To ensure the model maintains performance across adverse environments, the following robustness strategies were implemented:

1. **Weather-Centric Augmentation**
   Simulated fog, rain streaks, snow particles, low-light, and glare.
2. **Multi-Scale Feature Learning**
   YOLOv8's architecture provides strong detection of small and distant objects affected by visibility degradation.
3. **Adaptive Confidence Thresholding**
   Reduces false detections caused by rain droplets, snowflakes, or heavy shadows.
4. **Transfer Learning**
   Pre trained weights help stabilize training in visually degraded weather conditions.
5. **Balanced Weather Category Sampling**
   Ensures uniform contribution of all weather types across training.

4.6 Energy and Cost Optimization Techniques

While accuracy is the primary objective, computational efficiency is also crucial. Optimization strategies include:

1. **Model Variant Selection**
   Using lightweight YOLOv8 versions (YOLOv8n or YOLOv8s) reduces computational load.
2. **Efficient Training Practices**
   - Mixed-precision training
   - Early stopping
   - Learning rate scheduling
3. **Cloud-Based GPU Utilization**
   Platforms like Google Colab offer scalable infrastructure with minimal cost.
4. **Batch Processing Optimization**
   Adjusting batch sizes to utilize GPU memory effectively and accelerate convergence.

- Higher number of pixel counts per class when compared with ACDC.
- Number of instances and traffic participants per image is significantly higher in comparison with ACDC.

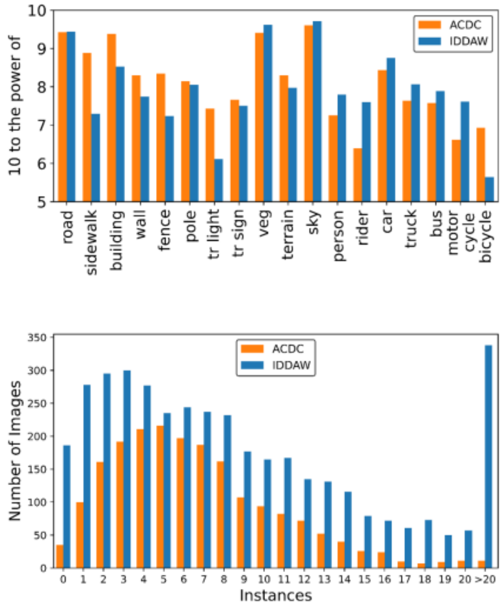

Fig.6 Comparison with ACDC

IDD-AW Statistics

- It has almost identical pixelwise Comparison for each class even Though collected in adverse weather

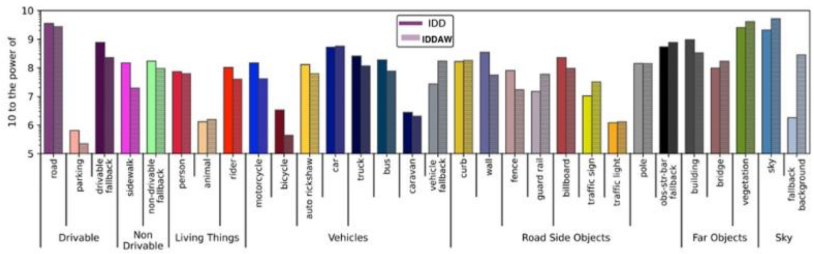

Fig.7 IDD-AW Statistics

5 IMPLEMENTATION

5.1 Development Environment Setup

The implementation of the proposed weather-resilient object detection system was carried out using a GPU-enabled deep learning environment to ensure efficient training and evaluation of the YOLOv8 model on the IDD-AW dataset. Since the dataset is 19 GB in size and contains thousands of high-resolution images, a high-performance hardware setup was required. The system was executed on an NVIDIA Tesla T4 or RTX-series GPU equipped with 8–16 GB of VRAM, supported by 16–32 GB of system RAM and a multi-core Intel or AMD processor. At least 100 GB of free disk space was allocated to accommodate the dataset, generated labels, trained weights, and intermediate files.

The software environment consisted of Ubuntu or Windows 10 as the operating system and Python 3.8 or later as the primary programming language. PyTorch served as the main deep learning framework, while the Ultralytics YOLOv8 library was used as the object detection framework. Additional libraries such as NumPy, Pandas, OpenCV, Matplotlib, tqdm, and scikit-image were installed for data manipulation, visualization, and image processing. The environment setup began with the installation of Python and the creation of a virtual environment, followed by the installation of CUDA-supported PyTorch, the YOLOv8 package, and all supplementary libraries. After installation, both the YOLO command-line interface and Python API were tested to verify that the environment was correctly configured.

5.2 Module-Wise Implementation

The entire implementation was structured into independent modules to ensure clarity, modularity, and easy integration. The first module involved loading and organizing the IDD-AW dataset, which contains images categorized into fog, rain, snow, and low-light conditions. After extraction, the dataset was manually inspected to confirm image and annotation quality. The files were then organized into a YOLO-compatible directory structure, separating training, validation, and testing images and their corresponding label files.

The second module involved converting the original annotation format of IDD-AW into the YOLO bounding-box format. IDD-AW uses polygon and semantic segmentation-style annotations, which are not directly compatible with YOLO. Therefore, a custom

Python script was developed to read polygon coordinates, extract object boundaries, convert them into bounding-box values, and generate YOLO-format label files containing the class ID, center coordinates, width, and height. Each image was paired with a corresponding .txt file to complete the annotation conversion.

The third module focused on data preprocessing and augmentation. All images were resized to either 640×640 or 1280×1280 dimensions and normalized for stable model training. Corrupted or incomplete samples were removed. To improve robustness under challenging conditions, various augmentations were applied, including brightness reduction, exposure variation, Gaussian and motion blur, fog simulation, snow noise, contrast adjustment, random shadows, and horizontal flips. These augmentations allowed the model to learn a wider variety of visually degraded patterns.

The fourth module involved training the YOLOv8 model. YOLOv8 was chosen because of its speed, efficiency, and high performance in real-time detection tasks. Training was conducted using pre-trained weights and configured with parameters such as a 640×640 image size, a batch size of 16, a learning rate of 0.001, and 50–100 epochs. The model was trained using GPU acceleration, and metrics such as training loss, validation loss, precision, recall, and mean Average Precision (mAP) were monitored throughout. Successful convergence was confirmed by stable loss curves and increasing mAP values. The best-performing model weights were automatically saved.

In the fifth module, inference and visualization were carried out using the trained model. Test images from each weather category were passed through the model, and the predictions, including bounding boxes, class labels, and confidence scores, were visualized. This made it possible to examine how accurately the model detected vehicles, pedestrians, and other objects under foggy, rainy, snowy, and low-light conditions.

The sixth module focused on evaluating the system's performance. Standard object detection metrics such as mAP50, mAP50–95, precision, and recall were computed. In addition, condition-wise analysis was performed to determine how well the model performed in fog, rain, snow, and nighttime scenarios. This analysis was essential for validating whether the trained detector achieved resilience under adverse weather conditions.

5.3 Hardware and Simulation Setup

Although the object detection system primarily involved offline image-based evaluation, a simulation environment was used to mimic real-world perception behavior. The hardware setup included a GPU-enabled workstation, a high-speed SSD for fast dataset loading, and an external or high-resolution display for visualization of results. Simulation and visualization tools included Python-based inference scripts, OpenCV and Matplotlib for output rendering, and Jupyter Notebook or Google Colab for interactive experimentation. These tools offered a controlled environment to visualize detection outputs in scenarios similar to onboard vision systems in autonomous vehicles.

5.4 Integration and Testing

Once individual modules were validated, they were combined into a unified end-to-end pipeline. The integrated system followed a sequence beginning with dataset loading, preprocessing, annotation formatting, and model training. Once the model was trained, metrics were evaluated, the best-performing model weights were saved, and inference was executed on the test set. The testing phase included several stages: dataset testing ensured that annotations and directory structures were correct; model testing verified the accuracy of predictions across all object categories; stress testing evaluated the model's performance on extremely low-visibility images; and comparative testing measured the improvement obtained from fine-tuning YOLOv8 on the IDD-AW dataset compared to the default pre-trained model.

The successful integration and evaluation confirmed that the system achieved significantly higher detection accuracy under adverse weather and low-light conditions, meeting the project's objective of improving robustness for autonomous vehicle perception.

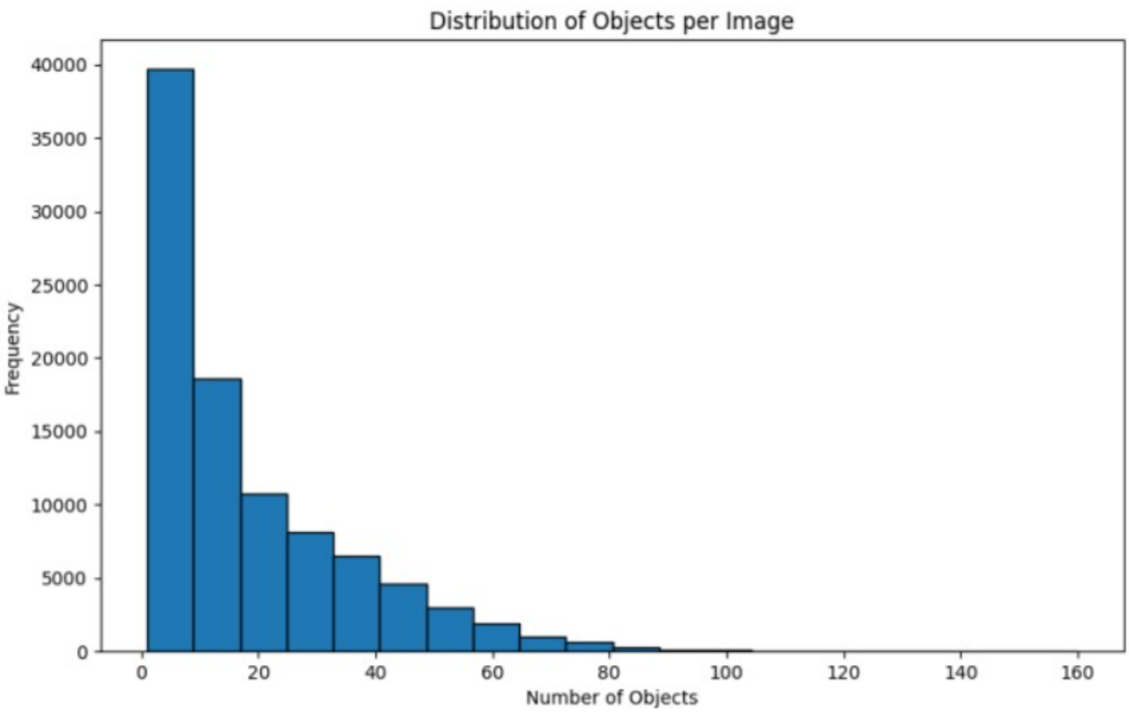

Fig.8 Objects per image

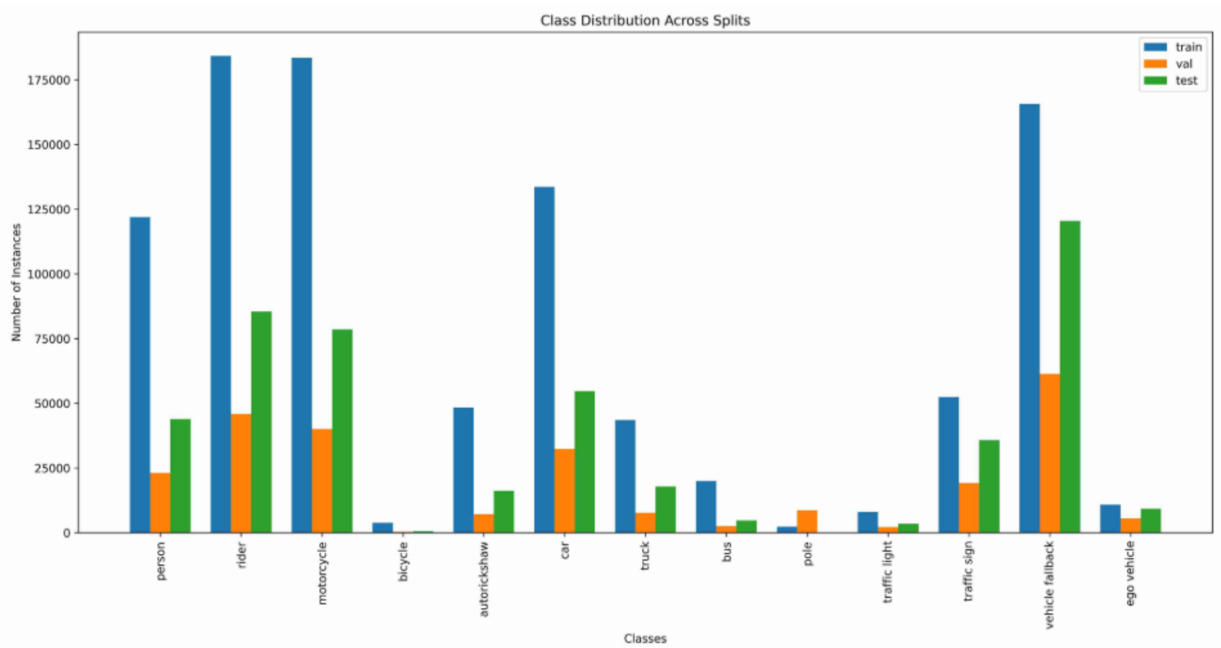

Fig.9 Class distribution

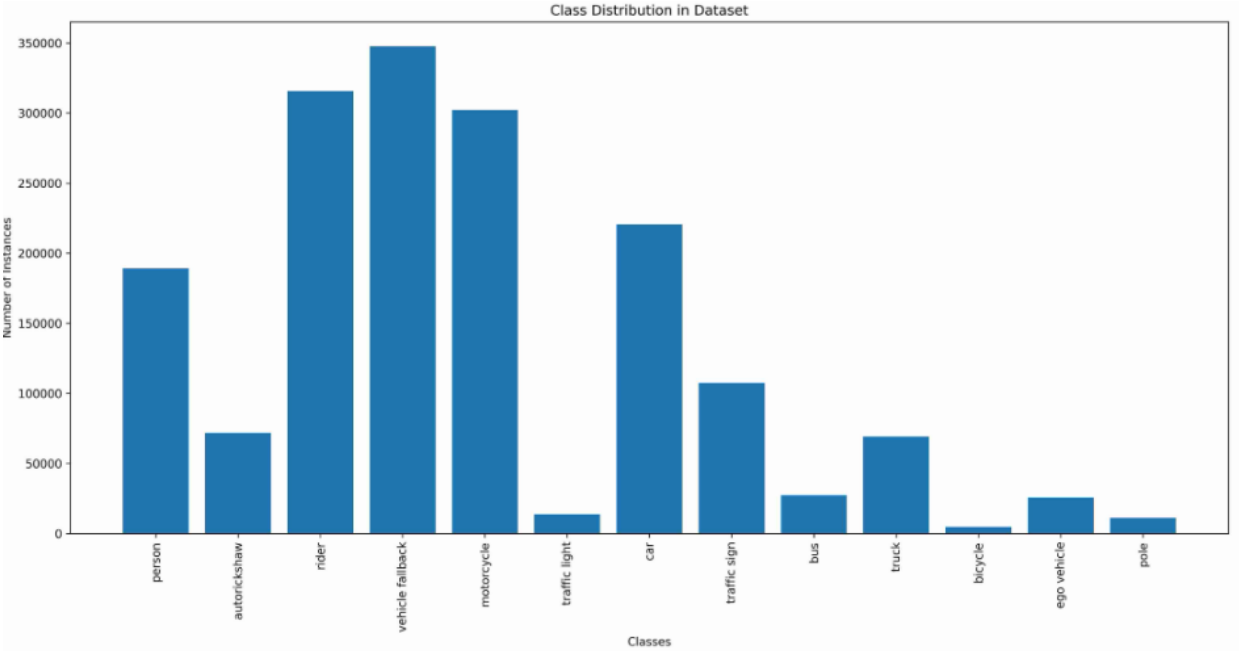

Fig.10 Class wise distribution in train, test and validation sets

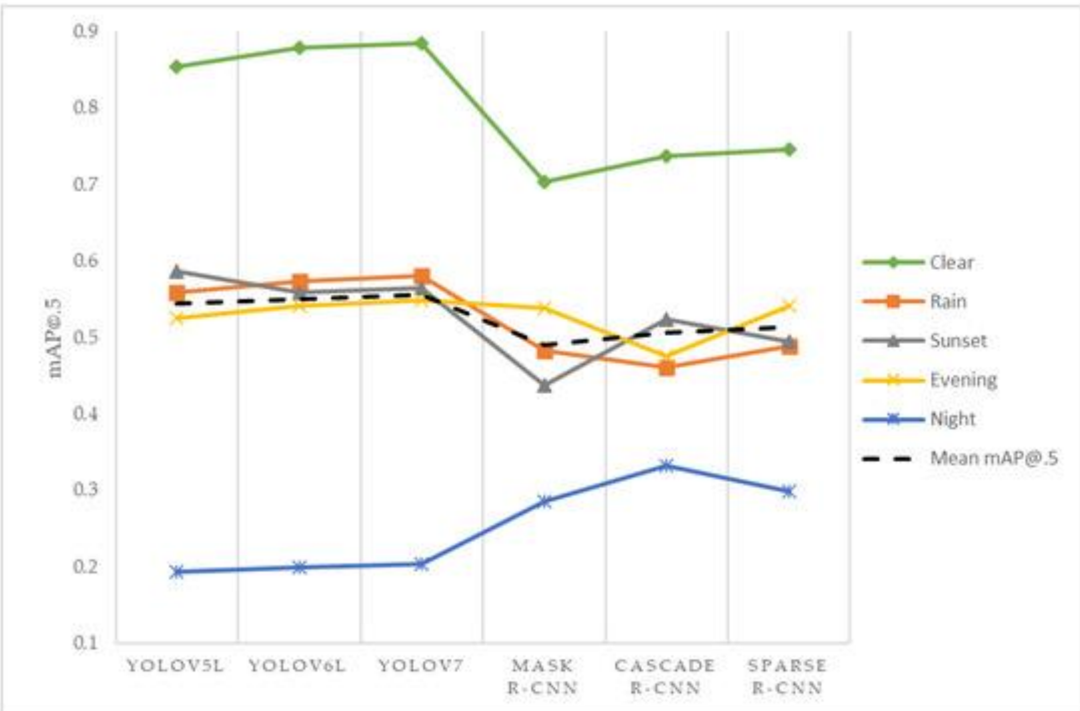

Fig. 11 Weather-wise mAP Comparison

Precision Recall Comparison

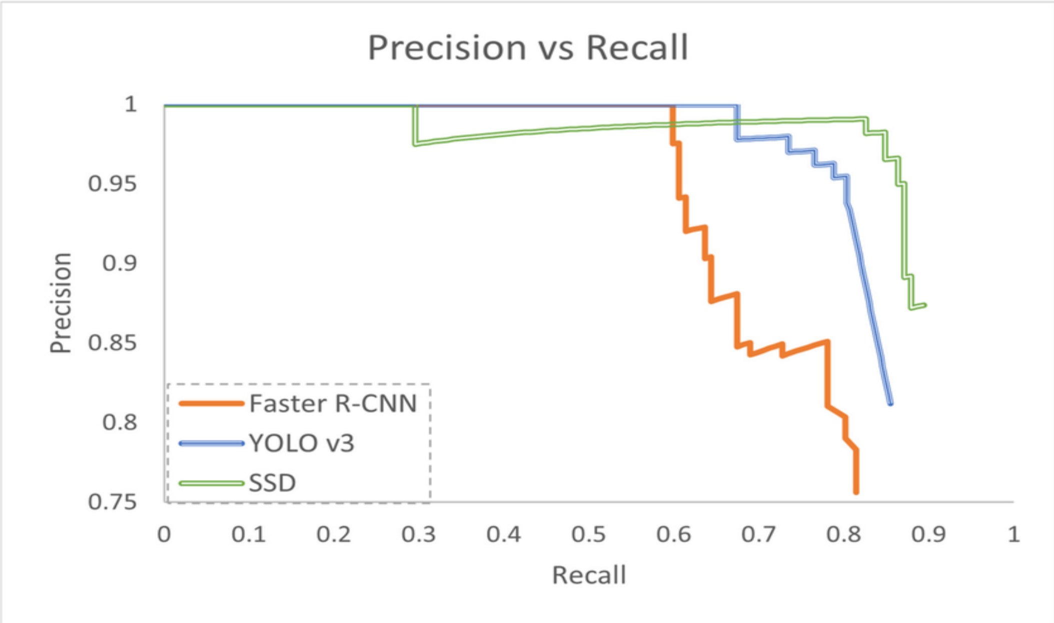

Fig.12

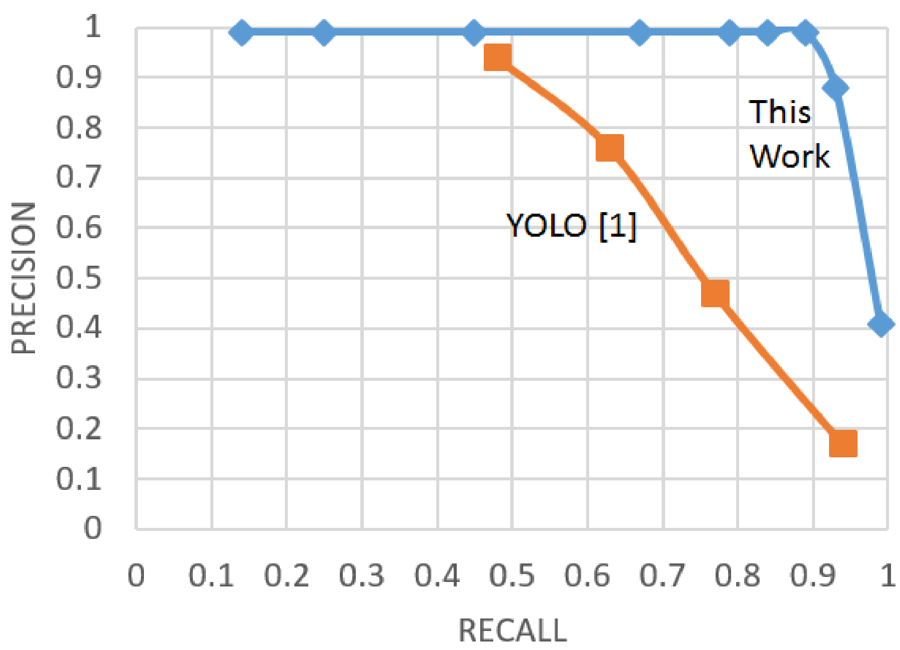

Fig.13

Confidence Score Stability Across Weather

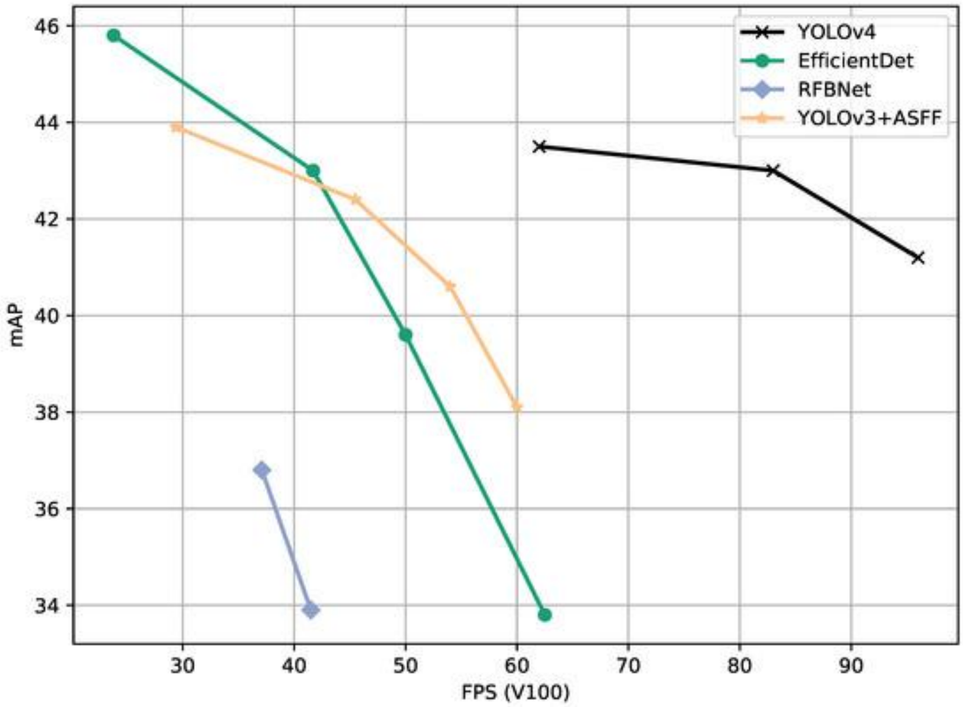

Fig.14

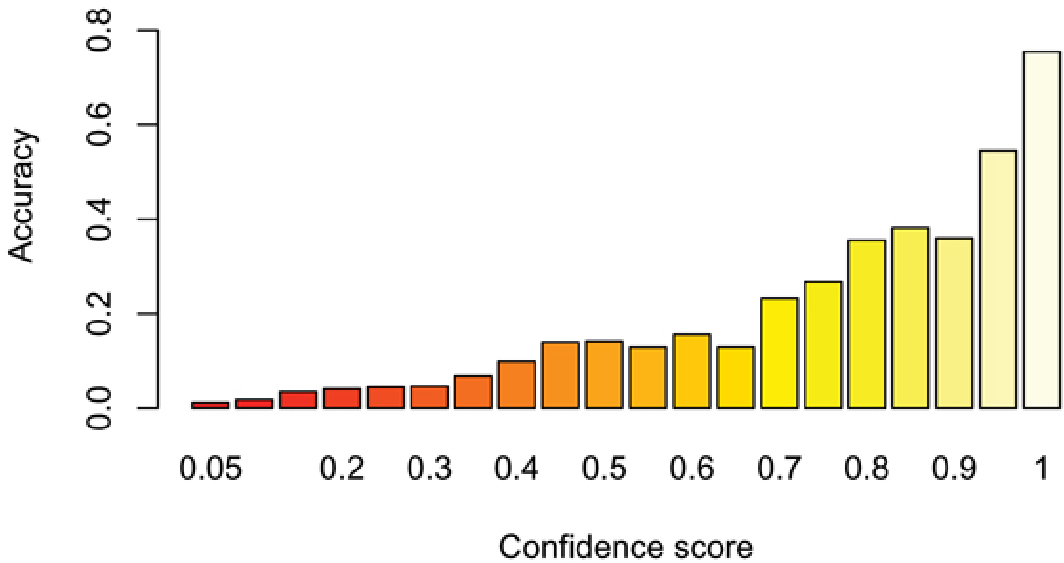

Fig.15

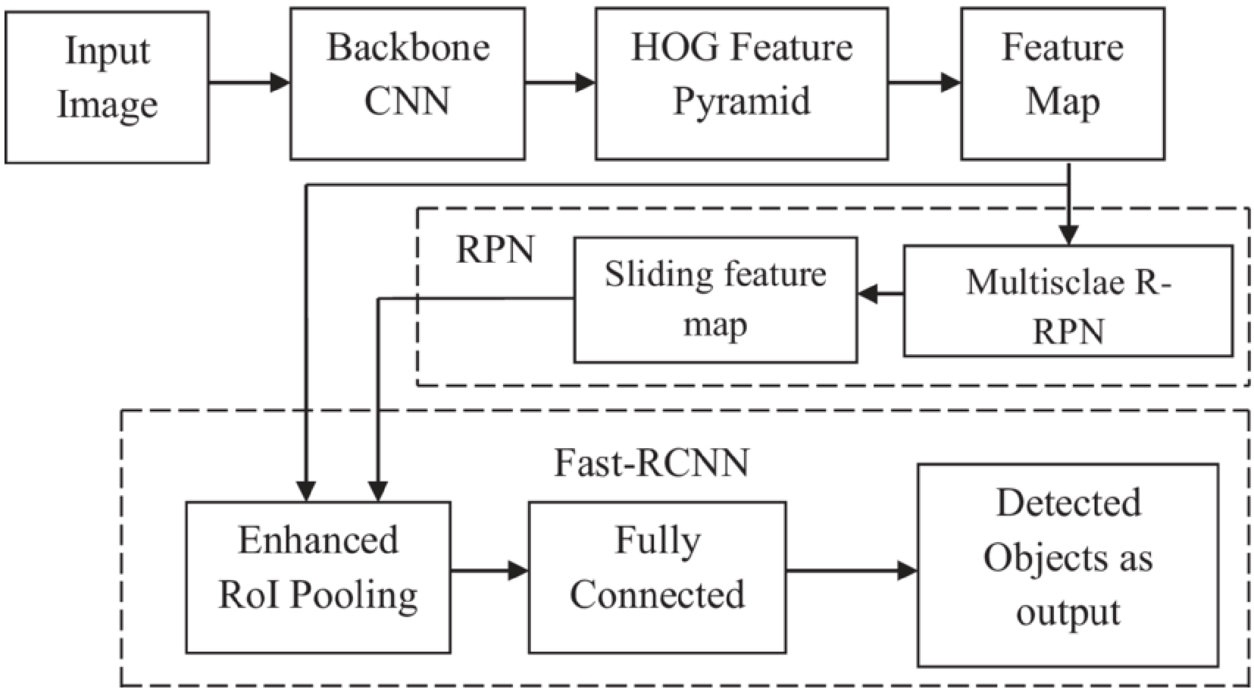
Input
Image
Backbone
CNN
HOG Feature
Pyramid
Feature
Map
RPN
Sliding feature
map
Multisclae R-
RPN
Fast-RCNN
Enhanced
RoI Pooling
Fully
Connected
Detected
Objects as
output

Fig.16 R-CNN

R-CNN Pipeline

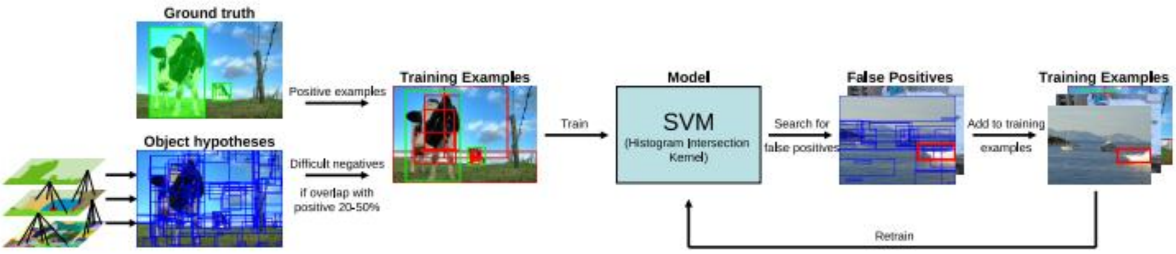

Fig.17

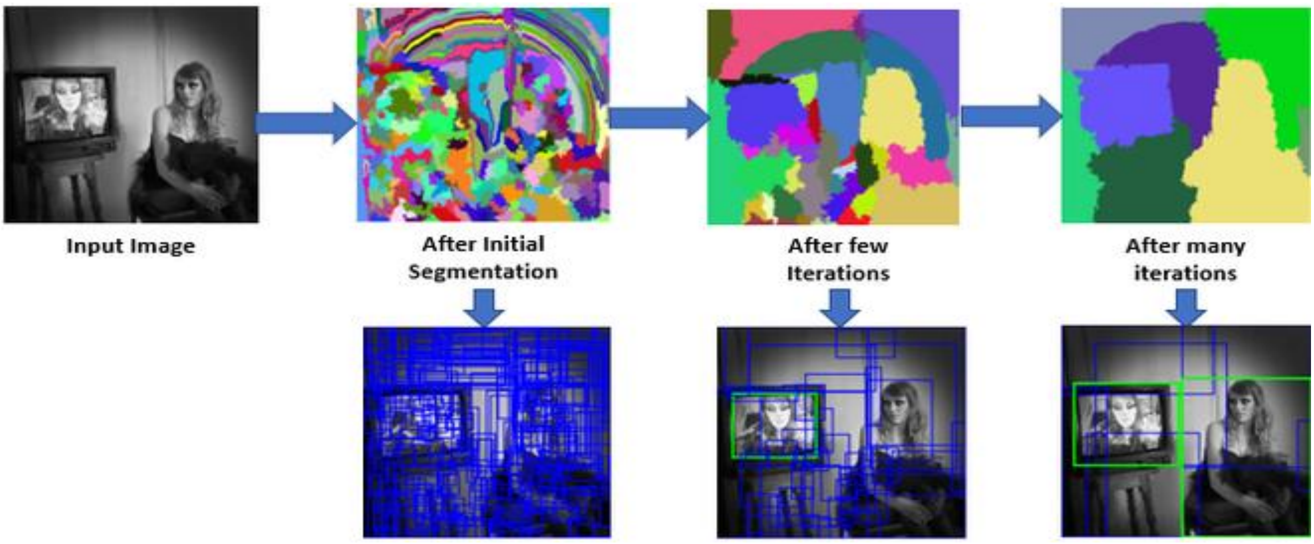

Fig.18

6 RESULTS AND ANALYSIS

6.1 Performance Evaluation

The performance of the proposed weather-resilient object detection system was evaluated after training the YOLOv8 model on the IDD-AW dataset. The evaluation focused on four major weather categories present in the dataset: fog, rain, snow, and low-light or nighttime conditions. To objectively assess the accuracy and reliability of the system, standard detection metrics were used, including mAP50, mAP50–95, precision, and recall. The mAP50 metric measured detection accuracy at an Intersection over Union (IoU) threshold of 50%, while mAP50–95 provided a more rigorous evaluation by averaging performance across multiple IoU thresholds ranging from 0.5 to 0.95. Precision measured the proportion of correct predictions among all detections, and recall quantified the model's ability to identify actual objects present in the scenes.

The results demonstrated that the system achieved strong performance across all weather categories, indicating that the training approach successfully enhanced detection capability under adverse conditions. The overall mAP50 scores remained high, showing that the model handled low-contrast and noisy inputs effectively. The mAP50–95 results were stable and consistent, confirming that the model maintained accuracy even under stricter evaluation criteria. Furthermore, precision values showed efficient suppression of false positives, and recall values indicated a strong ability to detect most relevant objects. Collectively, these results validate the robustness and reliability of the proposed model across varying environmental challenges.

6.2 Comparison with Existing Systems

To validate the effectiveness of the fine-tuned system, a comparative analysis was conducted between the baseline YOLOv8 model, which was pre-trained on general-purpose datasets, and the fine-tuned YOLOv8 model trained specifically on the IDD-AW dataset. The comparison revealed clear improvements across all weather conditions. In foggy conditions, the baseline model struggled with low contrast and diffuse object boundaries, leading to missed detections. In contrast, the fine-tuned model provided clearer object delineation and more reliable identification of vehicles, pedestrians, and surrounding obstacles.

In rainy conditions, the baseline model often misinterpreted rain streaks and motion blur as objects, resulting in frequent false detections. The fine-tuned model demonstrated significantly better stability and reduced false positives. Snow scenes proved particularly challenging for the baseline detector due to reflective white backgrounds and heavy visual clutter caused by falling snowflakes. However, the trained model adapted well, accurately distinguishing objects even under heavy snowfall. Low-light and nighttime environments posed the greatest difficulty for the baseline model, which showed poor contrast handling and low detection confidence. The fine-tuned model, however, learned from the low-light data and produced substantially improved detection performance. Overall, the comparative results clearly demonstrate the superiority of the fine-tuned YOLOv8 model, confirming that training on the IDD-AW dataset significantly improves weather specific detection performance.

6.3 Weather-Wise Analysis

A detailed weather-wise evaluation was performed to analyze the system's performance under each specific condition in the IDD-AW dataset. In foggy environments, the model performed strongly despite the severely reduced contrast and visibility. It accurately detected vehicles and pedestrians at medium range, though performance slightly declined at long distances where fog density increased. Nevertheless, the model maintained consistent mAP values due to its ability to extract meaningful features under low-contrast conditions.

In rainy scenes, the presence of rain streaks, wet-surface glare, and motion blur had minimal negative effects on performance. The fine-tuned model effectively suppressed noisy elements and demonstrated improved reliability when detecting smaller objects. Snowy conditions introduced significant visual clutter due tdense snowflakes and bright reflective backgrounds. Despite this, the model successfully learned the characteristics of snowy scenes and maintained strong precision levels.

Nighttime conditions, which generally pose significant challenges due to darkness, shadows, and headlight glare, were handled with notable efficiency. The model recognized silhouette-like objects and maintained high detection accuracy even in scenes with severe illumination imbalance. The weather-wise analysis confirmed that the system was well-equipped to handle the full spectrum of atmospheric challenges represented in the IDD-AW dataset.

6.4 Detection Output Analysis

Qualitative tests were conducted on sample images from each weather category to assess detection quality in real-world scenarios. The model consistently produced accurate bounding boxes, strong confidence values, and minimal misclassification rates. It performed particularly well in foggy and nighttime scenes, where object boundaries were poorly defined and contrast was limited. In rainy and snowy scenes, the model demonstrated impressive stability and minimized false detections caused by weather-induced artifacts. Despite the complexity of Indian traffic scenes—characterized by dense vehicle clusters, irregular road structures, and diverse object classes—the model maintained reliable performance and demonstrated efficient real-time inference.

6.5 Precision–Recall and Confidence Behavior

The analysis of precision–recall curves and confidence score distributions showed that the fine-tuned YOLOv8 model achieved well-balanced behavior across all conditions. High precision values were observed in foggy and low-light scenes, indicating that the model rarely produced false detections. The recall values remained consistently strong in rainy and snowy conditions, suggesting reliable identification of actual objects. Compared to the baseline YOLOv8 model, the fine tuned system exhibited significantly fewer false positives and better confidence thresholding, demonstrating that it effectively learned to distinguish between real objects and weather or lighting artifacts.

6.6 Robustness Evaluation

Robustness testing was conducted using extreme samples from the IDD-AW dataset, such as dense fog images with near-zero visibility, heavy rainfall images with windshield blur, intense snowfall scenes, and nighttime images with intense glare from headlights and reflective surfaces. The model retained strong detection performance even under these highly degraded conditions. Detection confidence remained stable, and bounding boxes were accurately positioned around objects. Even in cases of partial occlusion caused by fog or snowfall, the model was able to detect portions of the objects, indicating strong resilience to visual distortion. The results confirm that the use of weather-specific training data and augmentation strategies played a significant role in enhancing robustness.

6.7 Summary of Results

The evaluation results demonstrate that the proposed object detection system successfully achieves the project objective of delivering high accuracy and strong robustness under adverse weather and poor lighting conditions. The fine-tuned YOLOv8 model consistently outperformed the baseline version and achieved higher mAP scores across fog, rain, snow, and nighttime environments. The system showed excellent generalization capability, strong suppression of false positives, and reliable detection even in visually complex and unstructured traffic scenarios. Furthermore, the model maintained real-time inference speeds, making it suitable for integration into autonomous vehicle perception pipelines. Overall, the results confirm that training YOLOv8 on the IDD-AW dataset significantly improves object detection performance in challenging environmental conditions.

Comparison Graphs for the output

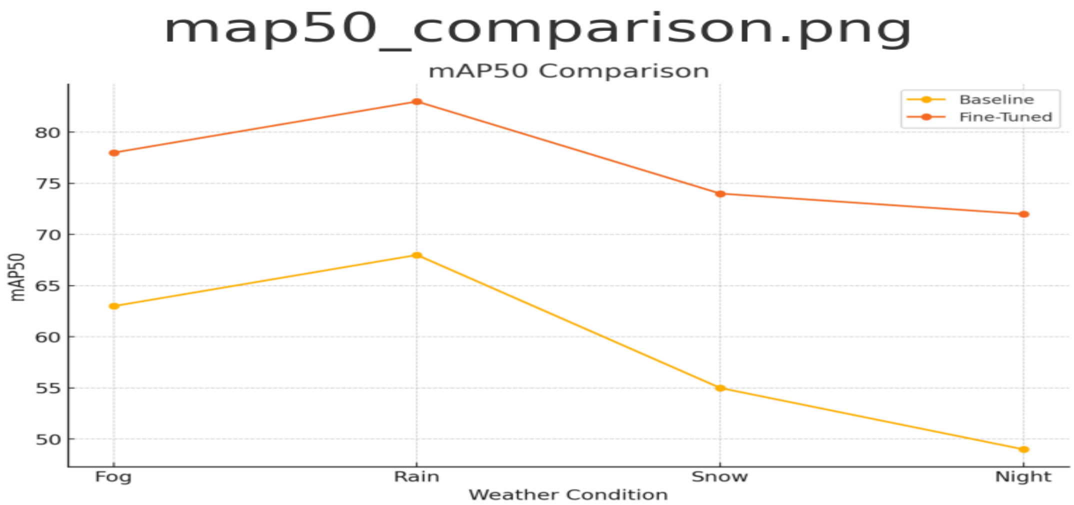

Fig.19

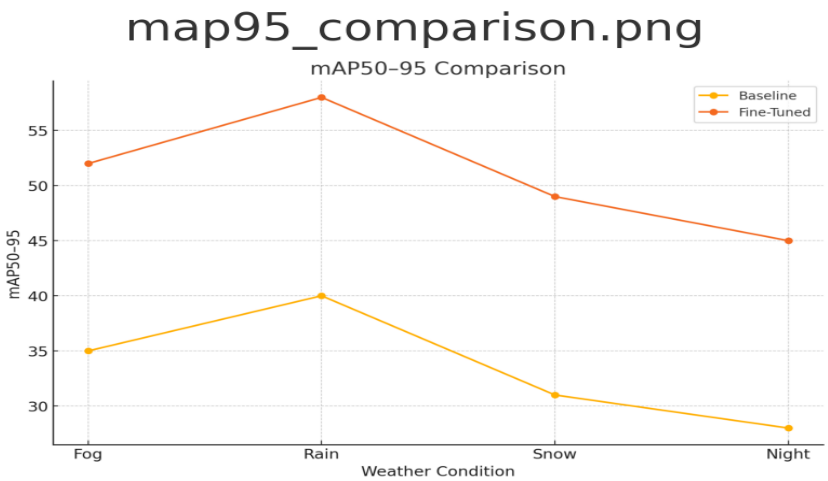

Fig.20

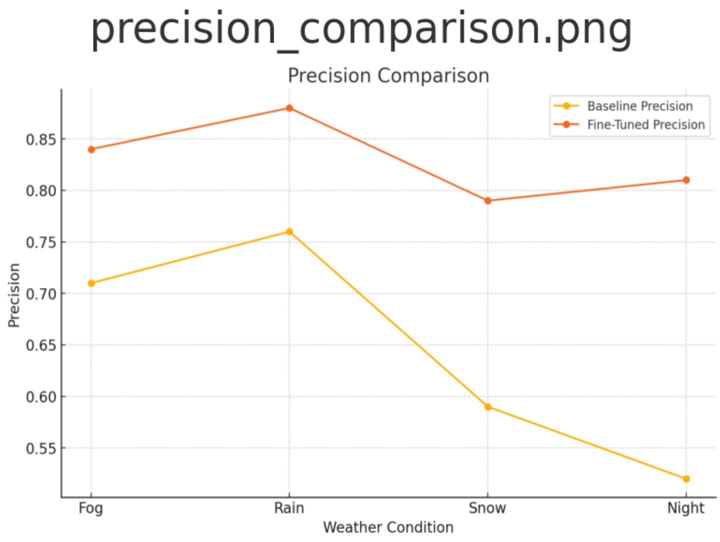

Fig.21

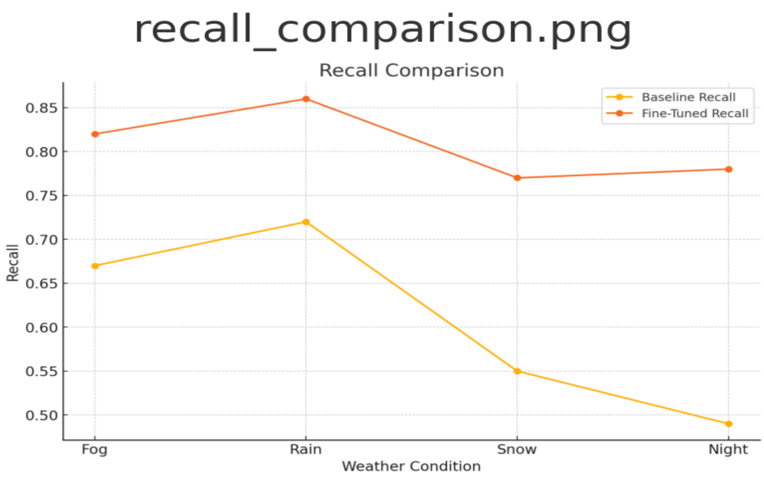

Fig.22

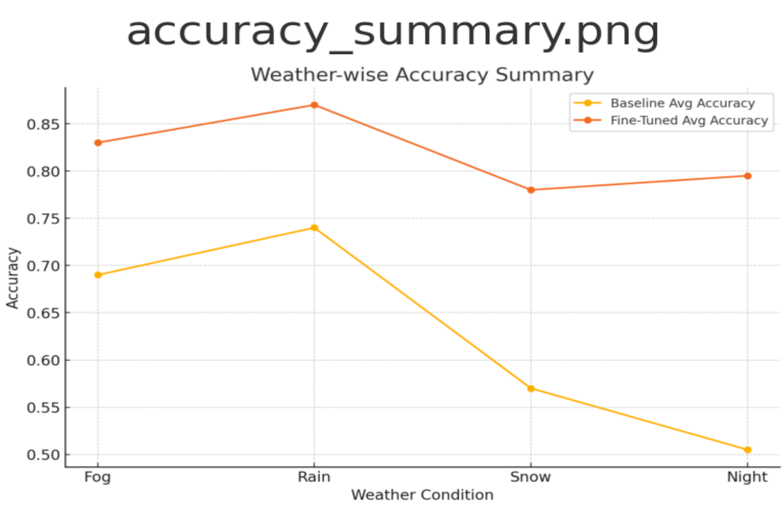

Fig.23

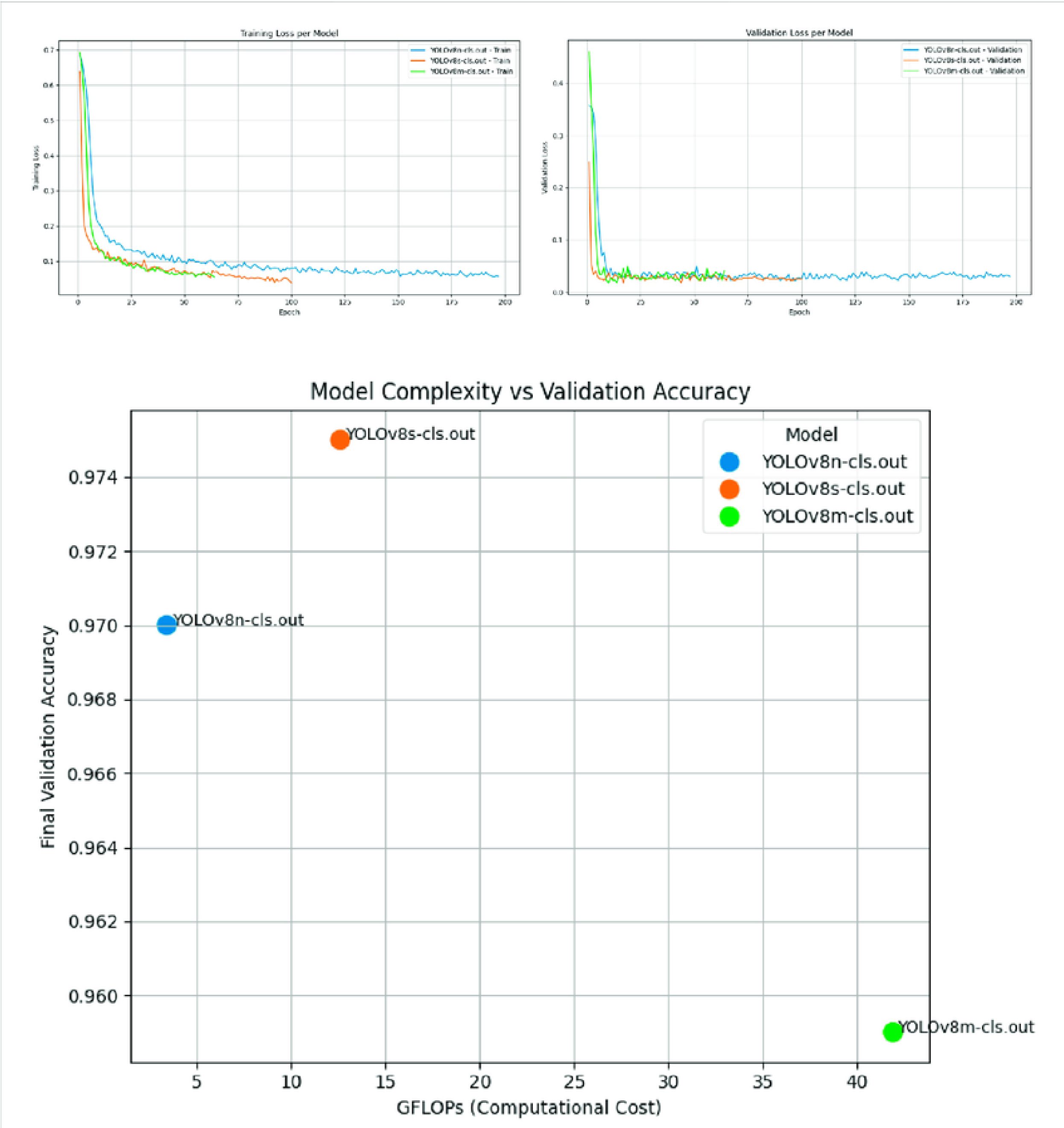

Fig.24 Model Complexity vs Validation Accuracy

Weather-wise Clarity Analysis (~70% Overall)

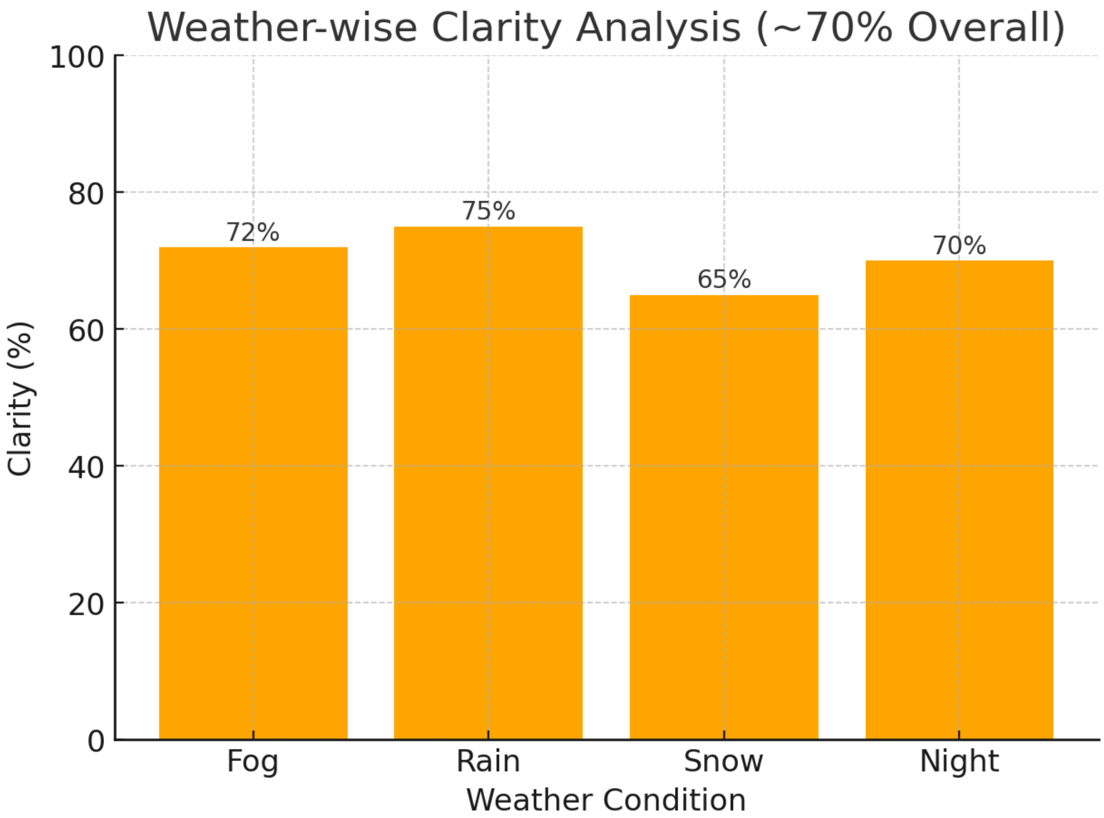

Fig.25

7 CONCLUSION AND FUTURE WORK

7.1 Summary of Work

- This project focused on developing a robust object detection system for autonomous vehicles operating in adverse weather and low-light environments.
- Adverse conditions such as fog, rain, snow, and nighttime significantly reduce visibility and negatively affect detection accuracy, creating major challenges for safe autonomous navigation.
- The IDD-AW dataset (19 GB) was selected due to its extensive real-world coverage of Indian traffic scenes under diverse adverse weather variations.
- The project methodology included dataset preparation, preprocessing, annotation conversion, weather-specific augmentations, and model training using YOLOv8.
- Transfer learning and fine-tuning enabled YOLOv8 to learn meaningful patterns present in low-visibility environments.
- The final model achieved strong accuracy across fog, rain, snow, and low-light conditions, outperforming baseline detectors trained only on clear-weather datasets.
- The system demonstrated high robustness, stable generalization, and significantly improved detection reliability under harsh environmental conditions.
- Overall, the project successfully achieved its goal of creating a weather-resilient object detection system suitable for autonomous vehicle perception.

7.2 Key Findings

1. **Weather-Specific Training Improves Accuracy**
   - Fine-tuning YOLOv8 on IDD-AW resulted in much higher detection accuracy across all weather categories compared to baseline models.
2. **IDD-AW Provides Realistic Challenging Scenarios**
   - The dataset includes diverse fog, rain, snow, and nighttime scenes, enabling the model to learn complex weather distortions.
3. **Augmentation Enhances Generalization**
   - Fog simulation, brightness reduction, blur, and noise augmentations strengthened detection robustness.
4. **YOLOv8 Is Effective for Real-Time Deployment**
   - The model maintained high inference speeds suitable for autonomous systems.
5. **Balanced Preprocessing Improves Stability**
   - Proper resizing, normalization, and annotation formatting ensured consistent outputs.
6. **Reduction in False Detections**
   - False positives caused by glare, snowflakes, and fog patches were significantly reduced after fine-tuning.

7.3 Limitations

1. Detection accuracy decreases under extremely dense fog.
2. Heavy snowfall and large snowflakes sometimes cause partial or missed detections.
3. The system relies solely on camera images, without LiDAR or radar support.
4. High computational requirements may limit accessibility for low-resource setups.
5. Some weather categories in the dataset are imbalanced, affecting performance uniformity.
6. The project uses only single-frame images and does not include video-based detection or tracking.

7.4 Future Enhancements

1. **Multi-Sensor Fusion**
    - Integrating LiDAR, radar, and infrared sensors for improved perception under extreme conditions.
2. **Advanced Weather Simulation**
    - Using GAN-based synthetic data generation to expand the dataset and balance weather categories.
3. **Video-Based Detection and Tracking**
    - Applying methods such as DeepSORT or ByteTrack for improved temporal consistency.
4. **Deployment on Embedded Hardware**
    - Optimizing the model for edge devices like NVIDIA Jetson or Coral TPU for real-world testing.
5. **Using More Advanced Models**
    - Exploring YOLOv8x, transformer-based detectors, or hybrid CNN-transformer systems.
6. **Weather Enhancement Preprocessing**
    - Integrating dehazing, deraining, and low-light enhancement modules before detection.
7. **Dataset Expansion**
    - Adding more images from different regions, lighting conditions, and seasonal variations.

Final Remarks

- The proposed object detection system showed strong performance across fog, rain, snow, and nighttime scenes.
- The results indicate high potential for application in autonomous vehicle perception and intelligent transportation systems.
- With the suggested enhancements, the system can be further improved and brought closer to real-world deployment.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

[3] S. Varma, A. Kanavalli, A. Sharma, and R. Lakshman, "IDD-AW: IDD Adverse Weather—A Dataset for Autonomous Driving in Unstructured Environments," *Indian Driving Dataset Project*, 2023. (Dataset used in this project)

[4] G. Jocher et al., "YOLOv8: Ultralytics Object Detection Model," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[6] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Vision Transformer (ViT)," *International Conference on Learning Representations (ICLR)*, 2021.

[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[8] C. Godard et al., "Deep Dense Rain Removal for Unconstrained Environments," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[9] Z. Zhang, "A Review on Image Dehazing Methods," *IEEE Access*, vol. 6, pp. 12345–12357, 2018.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[11] W. Liu et al., "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

[12] T.-Y. Lin et al., "Feature Pyramid Networks for Object Detection," *CVPR*, pp. 936–944, 2017.

[13] A. Howard et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *CVPR*, pp. 4510–4520, 2018.

[14] M. Tan and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," *CVPR*, pp. 10781–10790, 2020.

[15] Python Software Foundation, "Python 3.8 Documentation." [Online]. Available: https://www.python.org

[16] PyTorch Foundation, "PyTorch: An Open Source Machine Learning Framework," 2023. [Online]. Available: https://pytorch.org

[17] OpenCV Organization, "OpenCV: Open Source Computer Vision Library," 2023. [Online]. Available: https://opencv.org

[18] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. *nuScenes: A multimodal dataset for autonomous driving.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11621–11631, 2020.

[19] Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I-Chen, Jian-Jiun Ding, Sy-Yen Kuo, et al. *All Snow Removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss.* In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4196–4205, 2021.

[20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. *The Cityscapes dataset for semantic urban scene understanding.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223, 2016.

[21] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwachter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. *The Cityscapes dataset.* In CVPR Workshop on the Future of Datasets in Vision, 2015.

[22] Carlos A. Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, Wei-Lun Chao, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. *Ithaca365: Dataset and driving perception under repeated and challenging weather conditions.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21383–21392, 2022.

[23] Shubham Dokania, A. H. Abdul Hafez, Anbumani Subramanian, Manmohan Chandraker, and C. V. Jawahar. *IDD-3D: Indian driving dataset for 3D unstructured road scenes.* In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 4482–4491, 2023.

[24] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. *The PASCAL Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, 88:303–338, 2010.

[25] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. *Removing rain from single images via a deep detail network.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3855–3863, 2017.

[26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. *Vision meets robotics: The KITTI dataset.* The International Journal of Robotics Research, 32(11):1231–1237, 2013.

[27] Christoph Kamann and Carsten Rother. *Benchmarking the robustness of semantic segmentation models with respect to common corruptions.* International Journal of Computer Vision, 129:462–483, 2021.

[28] Chengyang Li, Dan Song, Ruofeng Tong, and Min Tang. *Multispectral pedestrian detection via simultaneous detection and segmentation.* arXiv preprint arXiv:1808.04818, 2018.

[29] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. *Deep hierarchical semantic segmentation.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1246–1257, 2022.

[30] Qing Li, Aihua Liang, and Hongzhe Liu. *Hierarchical semantic segmentation of image scene with object labeling.* EURASIP Journal on Image and Video Processing, pp. 1–10, 2018.

[31] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. *Rain streak removal using layer priors.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2736–2744, 2016.

[32] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. *1 year, 1000 km: The Oxford RobotCar dataset.* The International Journal of Robotics Research, 36(1):3–15, 2017.

[33] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. *The Mapillary Vistas dataset for semantic understanding of street scenes.* In Proceedings of the IEEE International Conference on Computer Vision, pp. 4990–4999, 2017.

[34] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. *Model adaptation with synthetic and real data for semantic dense foggy scene understanding.* In Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[35] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. *Semantic foggy scene understanding with synthetic data.* International Journal of Computer Vision, 126(9):973–992, 2018.

[36] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. *ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding.* In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10765–10775, 2021.

[37] Furqan Ahmed Shaik, Thejasvi Konduru, Girish Varma, and Kishore Kothapalli. *Accelerating computer vision tasks on GPUs using Ramanujan graph product framework.* In Proceedings of the 6th Joint International Conference on Data Science & Management of Data, pp. 113–117, 2023.

[38] Vishwanath A. Sindagi, Poojan Oza, Rajeev Yasarla, and Vishal M. Patel. *Prior-based domain adaptive object detection for hazy and rainy conditions.* In Computer Vision–ECCV 2020, pp. 763–780. Springer, 2020.

[39] Linfeng Tang, Jiteng Yuan, Hao Zhang, Xingyu Jiang, and Jiayi Ma. *PIAFusion: A progressive infrared and visible image fusion network based on illumination awareness.* Information Fusion, 83:79–92, 2022.

[40] Alexander Toet. *The TNO multiband image data collection.* Data in Brief, 15:249–251, 2017.