International Journal of Sciences and Innovation Engineering
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: www.ijsci.com
Vol.02 No.07.
E-ISSN : 3049-0251
DOI: https://doi.org/10.70849/ijsci

Vision-Based Vehicle Detection in Adverse Weather: A Hybrid Approach using Dark Channel Prior and Lightweight CNNs for Enhanced Driver Safety

Raghav Swami[1], Tarun Agarwal[2], Riya Sharma[3], Raghav Dharwal[4]
[*1,2,3,4] *Department of Computer Science Engineering (AI), Rajasthan Technical University, Kota, India*

Article Info

Article History:

Published:06 July 2025

Publication Issue:
Volume 2, Issue 7
July-2025

Page Number:
82-93

Corresponding Author:
Raghav Dharwal

Abstract:

Driving in hazy and foggy conditions significantly impairs visibility, leading to a higher risk of severe multi-vehicle accidents. This paper proposes a real-time, vision-based driver assistance system that operates effectively without relying on expensive sensors like LiDAR or radar. Our solution employs a two-stage approach. First, the Dark Channel Prior (DCP) algorithm is used to computationally remove haze from video frames captured by a standard dashcam or smartphone. Second, a lightweight Convolutional Neural Network (CNN), specifically a fine-tuned YOLOv8n model, detects vehicles in the dehaze video stream. To optimize for speed, a critical factor in collision avoidance, we trained the model on grayscale versions of the dehazed images. This reduces the computational load by processing a single color channel instead of three, while retaining essential features for vehicle detection. The model was trained on a custom dataset focusing on vehicle orientation, achieving real-time performance of up to 12 frames per second (FPS). The final system provides timely alerts to drivers, significantly enhancing situational awareness and safety in low-visibility environments.

Keywords: Dark Channel Prior, CNN, YOLOv8n, Vehicle Detection, Haze Removal.

1. Introduction

1.1. Problem Statement

Adverse weather conditions, particularly fog and heavy haze, are a major contributing factor to traffic accidents worldwide. The reduction in visibility distance makes it extremely difficult for drivers to perceive other vehicles and obstacles, drastically reducing reaction times. A catastrophic example of this is the 200-vehicle pileup in central China, which occurred amid thick fog and resulted in a tragic loss of life. Such incidents highlight the critical need for robust driver assistance systems that can "see" through the haze and provide early warnings. While advanced sensors like radar and LiDAR can penetrate fog, their high cost prohibits widespread adoption in consumer vehicles. Therefore, developing an affordable, vision-only system that can reliably function in these conditions is a paramount challenge in automotive safety.

(a) (b)

Figure 1 - (a) 6 killed, 51 hurt in mishaps due to smog - The Tribune

(b) Dense fog in North India hits air, rail traffic; causes road accidents - Rediff.com

1.2. Motive

The main motivation for this study is to democratize the advanced characteristics of driver assistance. By leveraging ubiquitous and low-cost cameras, such as those integrated into vehicle dashboards or smartphones, we can provide life-saving technology to a broader range of drivers. The goal is to create a system that is not only effective but also accessible, running efficiently on consumer-grade hardware and providing a critical layer of safety to prevent accidents caused by poor visibility.

1.3. Proposed Solution

We propose a hybrid computer vision pipeline that tackles the challenge in two sequential steps. The system first ingests a live video feed and applies the **Dark Channel Prior (DCP)** algorithm to each frame. This computationally restores image clarity by estimating and removing the atmospheric haze. The resulting clear, dehazed video stream is then fed into a highly optimized, deep-learning model for real-time vehicle detection. The detection model, based on **YOLOv8n**, is trained to identify vehicles and their orientation, enabling an alarm system to alert the driver to potential collision threats ahead.

1.4. Novel Contributions

This work introduces several key innovations:

- **Haze-Specific Model Training:** Unlike conventional object detectors trained on clear-weather images, our CNN model is specifically trained on a dataset of images that have been processed with the DCP algorithm. This ensures the model is robust and specialized for the unique visual artifacts and characteristics of dehazed imagery.
- **Lightweight and Fast Architecture:** We prioritize processing speed by:
    1. Employing *YOLOv8n*, a lightweight yet powerful CNN architecture.
    2. Converting the dehazed images to grayscale before inference. This reduces the input data from three channels (RGB) to one, significantly cutting down computation time while preserving the salient edge and shape features necessary for vehicle detection.
    3. Exporting the final model to the **ONNX (Open Neural Network Exchange)** format for optimized, cross-platform inference.

- **Specialized Dataset:** The model was trained using the **Vehicle Orientation dataset**, which improves its ability to not just detect a vehicle but also understand its position relative to the camera, a crucial factor for threat assessment.

2. Methodology

This section provides a detailed account of the methods used to develop the vision-based driver assistance system. Our approach is a sequential pipeline that first clarifies the visual input using a classical algorithm and then analyzes it with a state-of-the-art neural network.

2.1. Data Acquisition and Preprocessing

The success of deep learning models essentially depends on the quality and relevance of the training data.

- **Dataset Selection**: We utilized the Vehicle Orientation dataset, which contains images of vehicles from various perspectives (front, rear, side).This is very important as vehicle alignment provides more context-related information than a simple restriction box. For instance, detecting a vehicle's rear indicates it is likely moving in the same direction, while detecting its front may signal an oncoming threat.
- **Preprocessing Pipeline**: To prepare the data for our specific problem, we implemented a two-step preprocessing pipeline. This pipeline was applied to every image in the original dataset to create the final training set.
    1. **Dehazing:** Each image was first processed using the Dark Channel Prior (DCP) algorithm (detailed in Section 2.2). This step simulates the real-world input the system would receive, forcing the model to learn from images that have been algorithmically cleared of haze.
    2. **Grayscale Conversion:** Following dehazing, the images were converted from a 3-channel RGB format to a single-channel grayscale format. This was a critical optimization decision. Visually, the primary features for identifying a vehicle—its silhouette, edges, corners, and general shape—are preserved in grayscale. Computationally, this reduces the input tensor size by two-thirds, drastically decreasing the number of parameters in the initial layers of the CNN and leading to a significant increase in processing speed (FPS).

2.2. Dark Channel Prior (DCP)

Dark Channel Prior (DCP) is an algorithm based on statistical observations without haze. It's used to estimate the thickness of haze in an image and reverse its effects.

The underlying model for a hazy image is expressed as:

$$I(x) = J(x) \cdot t(x) + A \cdot (1 - t(x))$$

Let's break down this equation:

- I(x) represents the pixel values of the hazy image that we observe (the input from the camera).

- J(x) is the scene radiance, which is the actual, clear image of the scene that we want to recover.
- A is the global atmospheric light. It represents the ambient light in the atmosphere, which is the primary cause of the color and brightness of the haze. It is typically assumed to be uniform across the scene.
- t(x) is the transmission map. This is a crucial component that defines how much of the original light from the scene (J(x)) is transmitted through the haze to the camera. A value of t(x) close to 1 means the object is near and little light is lost, while a value close to 0 means the object is far away and most of its light is scattered by haze.

The DCP algorithm works as follows:

1. **Estimate Atmospheric Light (A):** The algorithm first finds the "dark channel" of the hazy image. The dark channel is an image where each pixel's value is the minimum intensity found across all three color channels (R, G, B) within a local patch.The algorithm proceeds by detecting the most luminous pixels within the dark channel. These pixels are assumed to be the most heavily obscured by haze, and their intensity in the original hazy image I(x) provides a robust estimate for the atmospheric light A.

2. **Estimate Transmission Map (t(x)):** The transmission map is estimated from the haze imaging equation under the assumption of the dark channel prior. The core idea is that the intensity of the dark channel is directly related to the thickness of the haze.

3. **Recover Scene Radiance (J(x)):** using light A in the atmosphere and estimated transmission card t(x), the ambiguous equation is inverted to resolve the transparent image J(x):

$$J(x) = \frac{I(x) - A}{max(\tilde{t}(x), t_0)} + A$$ **Recover dehaze image**

Here, t0 is a small constant (e.g., 0.1) introduced to prevent the denominator from becoming zero and to preserve a small amount of haze for distant objects, which makes the resulting image look more natural. This equation is applied pixel by pixel to restore the entire image.

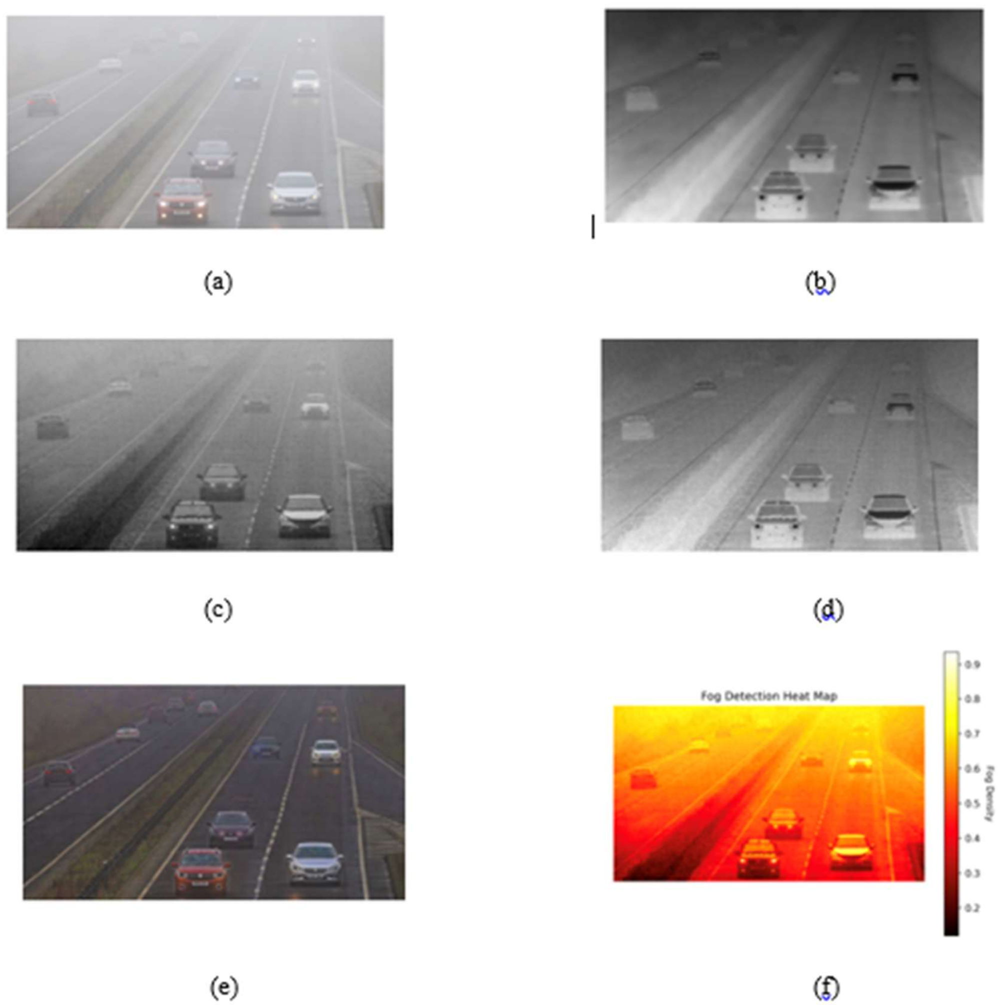


Figure 2 a) Original Image [2] b)Transmission Map c)Dark Channel Prior

d) Estimate Transmission e)Recovered Haze-Free Image f)fog Detected fog heat map

2.3. CNN Architecture and Training

For the essential task of detecting vehicles, we utilized a Convolutional Neural Network (CNN). CNNs are deep learning models specifically designed to analyze visual information.

What is CNN?

A CNN processes images through a series of specialized layers:

- **Convolutional Layers:** These are the foundational building blocks. They apply a set of learnable filters (or kernels) to the input image. Each filter is designed to detect a specific feature, such as an edge, a corner, a color blob, or more complex textures. As the image moves through successive convolutional layers, the filters progressively learn to recognize more complex features, such as a wheel, a window, and ultimately, a complete car.
- **Pooling Layers (or Subsampling):** These layers downsample the feature maps by decreasing their spatial dimensions (height and width). Max Pooling, a commonly used technique,

examines a small area and keeps only the maximum pixel value.This process improves computational efficiency and allows the model to remain unaffected by minor shifts or distortions in the input image.

- **Fully Connected Layers:** After several convolutional and pooling layers, the high-level features are flattened into a one-dimensional vector and fed into fully connected layers, which perform the final classification and localization tasks.

YOLOv8n Architecture

Our selection was *YOLOv8n ("nano")*, the most compact and efficient variant within the YOLO 11 series.YOLO (You Only Look Once) is renowned for its single-pass architecture, which makes it incredibly fast. Unlike older detectors that would look at an image multiple times, YOLO looks at the entire image just once to predict bounding boxes and class probabilities simultaneously. The architecture of YOLOv8n can be broken down into three main parts

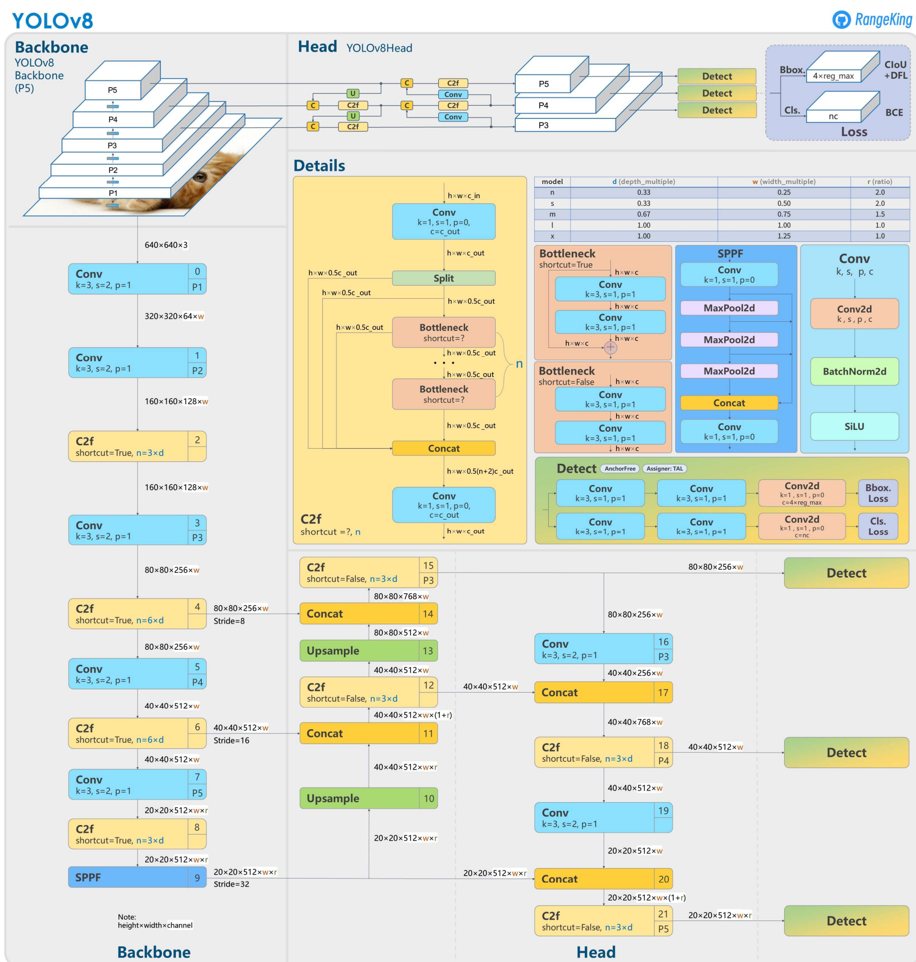


Figure 3 - The architecture of the YOLOv8 model, illustrating the Backbone, Neck, and Head components. [3]

1. **Backbone (CSPDarknet53):** This is the main body of the network. It's a deep CNN responsible for extracting rich features from the input image at various scales. YOLOv8 uses a

modified version of the CSP (Cross Stage Partial) Darknet architecture, which is highly efficient at feature extraction while minimizing computational cost.

2. **Neck (PANet):** The neck acts as a bridge between the backbone and the head. Its purpose is to aggregate and refine the features extracted by the backbone. YOLOv8 employs a **Path Aggregation Network (PANet).** PANet enhances the feature fusion process by adding a bottom-up path to the traditional top-down path of Feature Pyramid Networks (FPN). This allows high-level semantic features to be combined with low-level positional features, improving detection accuracy for objects of varying sizes.
3. **Head (Detection Head):** This is the final part of the network that performs the detection. The YOLOv8 head is **anchor-free**. Earlier detection models relied on fixed-size "anchor boxes" with various shapes to estimate object positions. An anchor-free head directly predicts the center of an object and its height and width, which simplifies the training process and can improve detection speed and performance. The head outputs a vector containing the bounding box coordinates, the confidence score (how certain the model is that an object is present), and the class probabilities (what type of object it is).

Training Details

- **Transfer Learning**: We did not train the YOLOv8n model from scratch. Rather than training from scratch, we utilized a model that had been pre-trained on a large-scale dataset such as COCO.This approach, known as **transfer learning**, leverages the general feature extraction capabilities the model has already learned (e.g., edges, textures, shapes). We then **fine-tuned** this pre-trained model on our specific dataset of dehazed, grayscale vehicle images. This significantly reduces training time and improves performance, especially with a specialized dataset.
- **Optimization and Export:** The training was conducted on a **P100 GPU** through Kaggle. After training, the model was converted into the **ONNX (Open Neural Network Exchange)** format, an open standard that supports interoperability across multiple platforms and frameworks. To optimize performance, the model was quantized to **FP16** (16-bit floating point), which significantly reduces its size and computational load. This optimization enables smooth execution on mobile devices by leveraging hardware acceleration through delegates such as Android's NNAPI. This streamlined pipeline allows the system to achieve approximately **12 FPS**, meeting real-time performance requirements.

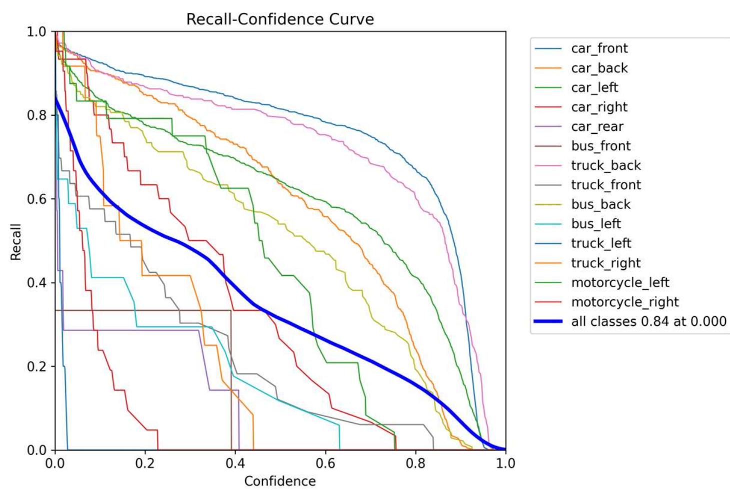
3. Results

The performance of our trained YOLOv8n model was evaluated on a held-out test set of dehazed, grayscale images. The model demonstrated strong performance in detecting vehicles under various simulated hazy conditions.

The key performance metrics are summarized below:

- **Precision:** The model demonstrated high precision, meaning its vehicle predictions were accurate the majority of the time. This level of accuracy is vital for an alarm system to reduce false alarms.

- **Recall:** The model also showed good recall, successfully identifying a high percentage of the actual vehicles present in the frames.
- **F1-Score:** The F1-score, representing the harmonic average of precision and recall, stayed consistently high across various confidence thresholds, indicating a well-balanced model
- **mAP50-95:** The mean Average Precision (mAP) over an IoU (Intersection over Union) range



of 0.5 to 0.95 was used as the primary metric for object detection accuracy.

Figure 4 - Recall-Confidence Curve for all object classes.

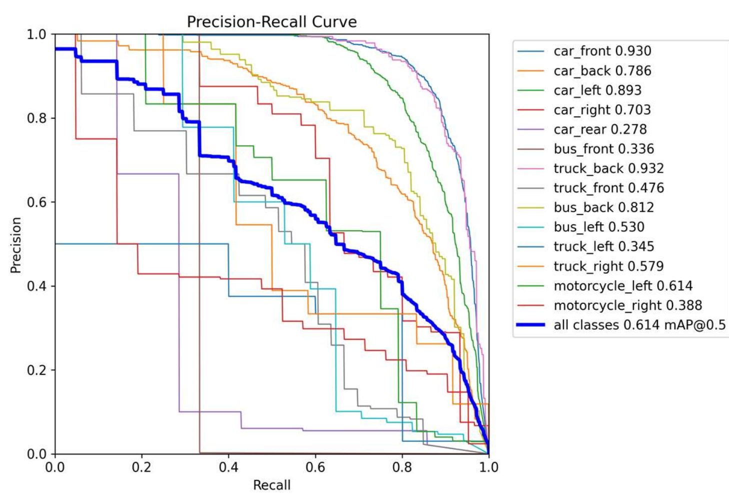


Figure 5 - The Precision-Recall Curve illustrates the balance between precision and recall.

Precision-Recall Curve, showing the trade-off between precision and recall.

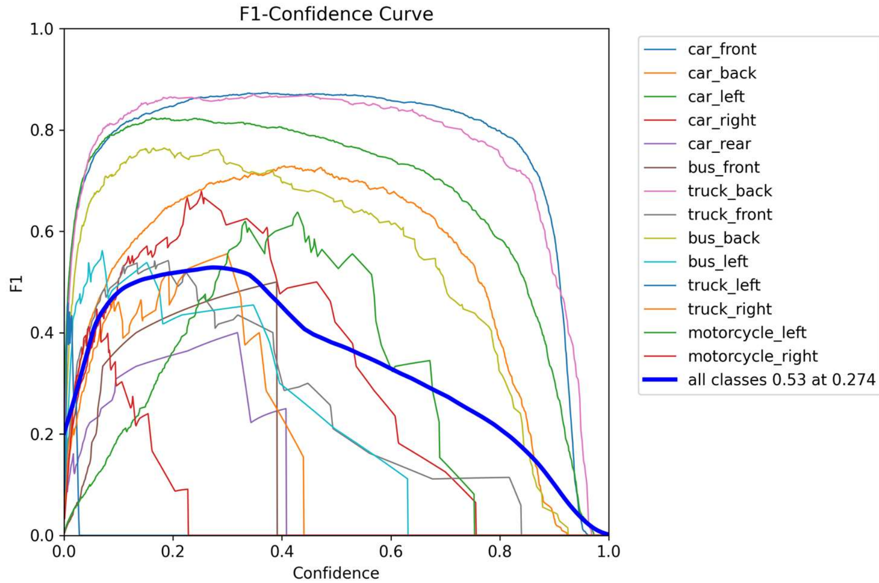


Figure 6 - F1-Score-Confidence Curve, indicating the model's balance at various confidence levels.

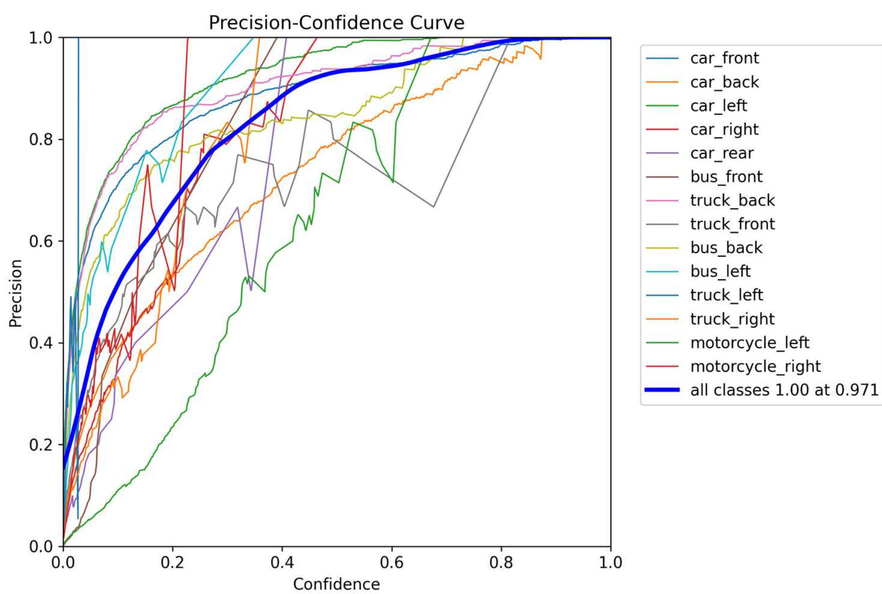
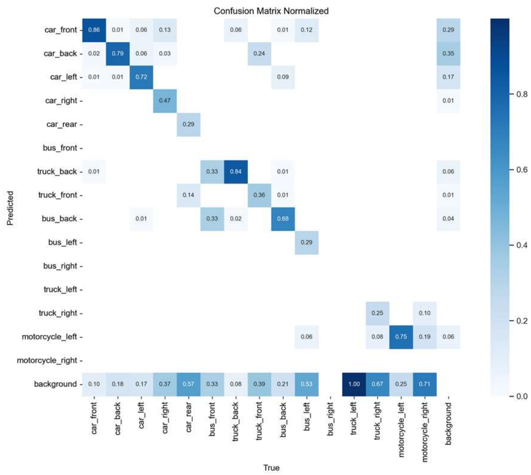


Figure 7 - Precision-Confidence Curve for all object classes.

The confusion matrices (both normalized and unnormalized) further illustrate the model's classification accuracy across different vehicle orientations. The strong diagonal presence indicates that the model is not only detecting vehicles but is also accurately classifying their orientation, which



adds valuable context for a driver assistance system.

Figure 8 - Normalized Confusion Matrix, showing the percentage of correct and incorrect predictions for each class.(.88=16889 instance)

4. Conclusion

This research successfully demonstrates the design and implementation of a real-time, vision-only driver assistance system capable of enhancing safety in adverse weather. By creating a hybrid pipeline that combines the **Dark Channel Prior (DCP)** algorithm for haze removal and a lightweight **YOLOv8n Convolutional Neural Network (CNN)** for vehicle detection, we have developed a promising and low-cost solution to the critical problem of reduced visibility.

The system effectively addresses the challenges of atmospheric haze and provides timely alerts about potential road threats, directly contributing to driver safety. Our findings confirm that this dual approach is not only viable but also efficient, achieving real-time performance on accessible hardware. This work validates that even without expensive sensors, significant safety improvements can be achieved by intelligently processing video streams from standard dashcams or smartphones, making advanced driver assistance more accessible.

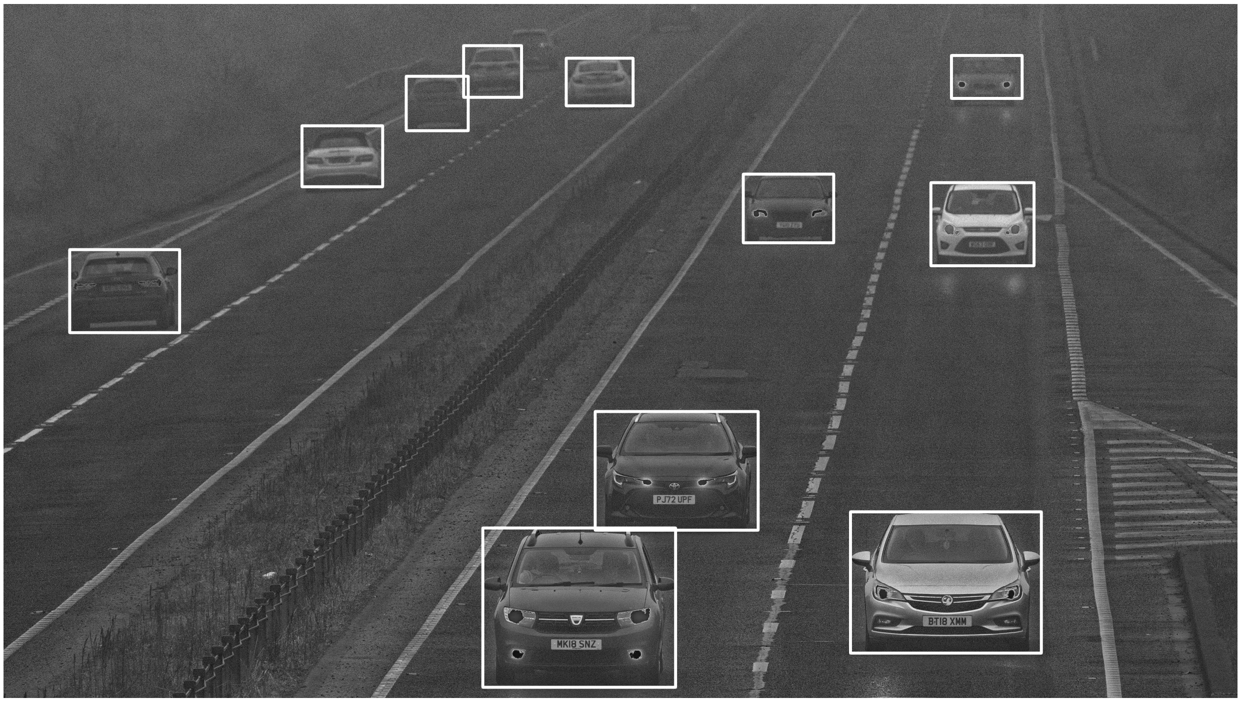


Figure 9:- Dehaze image with detection

5. Future Work

While our system provides a robust foundation, several avenues exist for future enhancement and research:

- **Advanced Dehazing Techniques:** Future iterations could explore moving beyond classical algorithms like DCP. Investigating and implementing state-of-the-art, **deep learning-based dehazing methods** could yield superior image restoration, particularly in dense and non-homogenous fog. These AI-powered models often provide a better trade-off between image quality and computational efficiency.
- **Further CNN Optimization:** The vehicle detection model can be further refined. This includes experimenting with different **lightweight CNN architectures** or applying more aggressive optimization techniques like advanced model quantization and pruning. The goal would be to further improve the **accuracy-speed trade-off**, pushing for higher frame rates and detection accuracy on dehazed video input.
- **Multi-Sensor Fusion:** To create an even more robust system, future work should focus on **integrating data from other common vehicle sensors**. Fusing the visual data from our system with inputs from an Inertial Measurement Unit (IMU), GPS, or even low-cost radar would create a comprehensive perception system. This would enhance reliability, reduce false positives, and ensure robust performance in even the most challenging weather and environmental conditions.

References

1. S. Chen, "Vehicle Orientation Dataset," GitHub, 2023. [Online]. Available: https://github.com/sekilab/VehicleOrientationDataset

2. UK weather: fog causes travel chaos with flights cancelled

3. YOLOv8: An Overview of the Key Architectural Enhancements by Rahima Khanam, Muhammad Hussain arXiv:2410.17725 **[cs.CV]** https://doi.org/10.48550/arXiv.2410.17725

4. K. He, J. Sun, and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341-2353, Dec. 2011.

5. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.

6. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft COCO: Common Objects in Context," in European Conference on Computer Vision (ECCV), Zurich, Switzerland, 2014.