

Article

A Real-Time Collision Warning System for Autonomous Vehicles Based on YOLOv8n and SGBM Stereo Vision

Shang-En Tsai *  and Chia-Han Hsieh

Department of Computer Science and Information Engineering, Chang Jung Christian University,
Tainan City 711, Taiwan

* Correspondence: sean@mail.cjcu.edu.tw; Tel.: +886-6-2785123 (ext. 6154)

Abstract

With the rapid development of autonomous vehicles and intelligent transportation systems, vehicle detection and distance estimation have become critical technologies for ensuring driving safety. However, real-world in-vehicle environments impose strict constraints on computational resources, making it impractical to deploy high-end GPUs. This implies that even highly accurate algorithms, if unable to run in real time on embedded platforms, cannot fully meet practical application demands. Although existing deep learning-based detection and stereo vision methods achieve state-of-the-art accuracy on public datasets, they often rely heavily on massive computational power and large-scale annotated data. Their high computational requirements and limited cross-scenario generalization capabilities restrict their feasibility in real-time vehicle-mounted applications. On the other hand, traditional algorithms such as Semi-Global Block Matching (SGBM) are advantageous in terms of computational efficiency and cross-scenario adaptability, but when used alone, their accuracy and robustness remain insufficient for safety-critical applications. Therefore, the motivation of this study is to develop a stereo vision-based collision warning system that achieves robustness, real-time performance, and computational efficiency. Our method is specifically designed for resource-constrained in-vehicle platforms, integrating a lightweight YOLOv8n detector with SGBM-based depth estimation. This approach enables real-time performance under limited resources, providing a more practical solution compared to conventional deep learning models and offering strong potential for real-world engineering applications.



Academic Editor: Sergio
Busquets-Monge

Received: 24 September 2025

Revised: 23 October 2025

Accepted: 29 October 2025

Published: 31 October 2025

Citation: Tsai, S.-E.; Hsieh, C.-H. A Real-Time Collision Warning System for Autonomous Vehicles Based on YOLOv8n and SGBM Stereo Vision. *Electronics* **2025**, *14*, 4275.

<https://doi.org/10.3390/electronics14214275>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: YOLOv8; object detection; Semi-Global Block Matching; collision warning system; Advanced Driver Assistance Systems (ADASs); real-time systems

1. Introduction

1.1. Pragmatic Challenges of Perception Technologies in Intelligent Transportation Systems

With the rapid development of Advanced Driver Assistance Systems (ADASs) and fully autonomous driving technologies, high-precision and highly reliable environmental perception has become the cornerstone of driving safety. Vision-based perception systems, particularly camera-based object detection and 3D localization, play a central role due to their low cost and rich information content. However, a widening “deployment gap” has emerged between the pursuit of algorithmic performance limits in academic research and the practical constraints faced in large-scale industrial deployment. At the heart of this gap lies a fundamental contradiction: state-of-the-art perception algorithms often come with massive model parameters and substantial computational demands, which

sharply conflict with the stringent power, cost, and computational limitations of in-vehicle embedded platforms.

In real-world vehicular environments, perception systems must complete detection, localization, and decision-making within milliseconds, without relying on high-end GPUs. Therefore, the value of an algorithm is determined not only by its accuracy on benchmark datasets but also by its real-time performance and efficiency on resource-constrained edge computing units. Recent research trends clearly reflect this shift, as both academia and industry have moved their focus from solely improving accuracy to developing more lightweight model architectures. For instance, models such as YOLOv8-Lite have been proposed to address the excessive parameter sizes and computational demands of conventional deep learning models. Their design philosophy emphasizes significantly reducing computational burden while maintaining high accuracy, thereby meeting the stringent requirements of real-world in-vehicle applications.

Consequently, the core objective of this study is not merely to propose a system with high accuracy, but to directly confront this deployment gap by presenting a pragmatic solution that achieves an optimal balance among computational efficiency, real-time performance, and deployment feasibility.

1.2. The Current State of Real-Time 2D Detection: Strategic Choice of CNNs over Transformers

In the evolution of object detection technologies, the YOLO (You Only Look Once) series has established itself as a benchmark in real-time applications owing to its remarkable balance between speed and accuracy [1,2]. From the first version, which reformulated detection as a regression problem, to subsequent versions that introduced anchor-based designs and optimized backbones, and finally to the latest YOLOv8 with its anchor-free design, efficient C2f modules, and decoupled detection heads, this series has continuously advanced in architecture, consistently pushing the performance ceiling of single-stage detectors [3,4].

In recent years, however, transformer-based detectors such as RT-DETR (Real-Time Detection Transformer) have gained significant attention in academia. Leveraging the global contextual understanding enabled by self-attention mechanisms, they have demonstrated superior accuracy in certain benchmark evaluations [5].

From a pragmatic deployment perspective, CNN-based YOLOv8 demonstrates stronger overall advantages. Several recent comparative studies indicate that although the largest RT-DETR models may slightly surpass YOLOv8 in mean Average Precision (mAP) on general datasets such as COCO, YOLOv8 significantly outperforms its counterparts in inference speed, particularly on central processing units (CPUs) [6]. For mass-produced vehicles that cannot be equipped with expensive GPUs, superior CPU performance is a decisive factor for real-world deployment. In addition, YOLOv8 is more efficient in terms of parameter count, computational complexity (GFLOPs), and training memory requirements, which translates into lower development and hardware costs [7–9].

A comparative study focusing specifically on ADAS scenarios directly confirmed this point. Using real driving datasets such as BDD100k, Table 1 concluded that YOLOv8 outperformed RT-DETR in both accuracy and speed, making it a more suitable choice for ADAS tasks [6]. This highlights an important phenomenon: a potential divergence between the architectural directions favored in academic research (e.g., transformers and other novel structures) and the practical demands of industrial deployment (favoring mature, efficient CNNs). Therefore, this study adopts YOLOv8n as the core detector not out of disregard for emerging technologies, but as a deliberate engineering decision grounded in data and application context, aimed at maximizing the system's practical value in real-world scenarios.

Table 1. YOLOv8 models demonstrate superior inference speed and efficiency, making them more suitable for real-time ADAS deployment, whereas RT-DETR achieves higher accuracy at the cost of significantly increased computational demand.

Model	mAP (COCO val)	GPU Inference Latency (T4, ms)	CPU Inference Latency (ONNX, ms)	Parameters (M)	GFLOPs	Key Advantages for ADAS Applications
YOLOv8n	37.3	2.22	45.9	3.2	8.7	Exceptional inference speed, ideal for the most demanding real-time scenarios
YOLOv8s	44.9	2.81	68.3	11.2	28.6	Excellent trade-off between speed and accuracy
RT-DETR-R18	46.5	7.40	-	21	69	Higher accuracy, but significantly increased GPU latency
RT-DETR-R50	53.1	10.8	-	43	129	Leading accuracy, but prohibitive computational cost, unsuitable for real-time deployment

Furthermore, the pursuit of efficiency in real-time detection has spurred the development of specialized lightweight architectures. Models such as YOLOv8-Lite and other compact CNN variants exemplify this trend by prioritizing lower computational cost and faster inference without a significant sacrifice in accuracy. These lightweight designs extend the applicability of real-time detectors to embedded platforms and edge devices, reinforcing the strategic rationale for selecting YOLOv8n in resource-constrained automotive environments.

1.3. Unresolved Frontiers in 3D Perception: The “Generalization Gap” in Stereo Depth Matchings

Two-dimensional detection alone is insufficient to support advanced driver assistance functions such as collision warning; the system must possess accurate three-dimensional perception capabilities, with depth estimation serving as a core component. Owing to its passive and low-cost nature, stereo vision has emerged as an attractive solution. Current stereo matching techniques can be broadly divided into two categories: traditional algorithms based on geometric and photometric consistency, and end-to-end deep learning networks. In parallel, monocular metric depth estimation has also been extensively studied [10], though it continues to suffer from issues such as scale ambiguity and reduced robustness in real-world deployments.

Deep learning methods, such as PSMNet and RAFT-Stereo [11], employ supervised training on large-scale synthetic datasets to construct highly nonlinear matching cost functions, achieving remarkable accuracy on public benchmarks. Several systematic reviews have also highlighted both the strengths and the persistent limitations of deep learning-based depth estimation, particularly its vulnerability to domain shift [12]. However, despite their apparent power, these models face a critical challenge in practical applications—the domain shift problem. Survey studies have explicitly pointed out that the lack of generalization to unseen scenarios remains one of the most pressing unresolved issues in deep stereo matching [11]. Models trained on synthetic datasets such as Scene Flow experience catastrophic performance degradation when directly applied to real-world scenarios like KITTI, where lighting conditions, textures, and sensor noise characteristics differ significantly [13]. Furthermore, studies have observed considerable fluctuations in generalization performance across different training epochs, implying that their deployed behavior is difficult to predict—an unacceptable uncertainty for safety-critical applications.

This dilemma challenges the conventional perception of algorithmic “robustness.” Deep learning is often assumed to be more resilient to noise and image variations than traditional approaches. Yet, in the specific and crucial dimension of domain generalization, the opposite holds true. Traditional algorithms, such as Semi-Global Block Matching (SGBM), operate based on local geometric and photometric constraints rather than learning from a specific data distribution. As a result, SGBM is inherently insensitive to dataset “domains,” maintaining consistent processing logic whether applied to synthetic imagery or real-world street scenes. This cross-domain consistency makes SGBM more reliable and robust than deep models when confronted with unpredictable real-world environments. For a collision warning system where safety is paramount, such predictable, cross-scenario robustness is far more valuable than achieving extreme accuracy on specific benchmarks.

1.4. Contributions of This Study: A Deployment-Oriented Pragmatic Hybrid Vision Framework

Building upon the above analysis, this study introduces a deployment-oriented hybrid vision framework that emphasizes system-level integration and validation rather than merely utilizing an existing library.

Unlike works that only apply or fine-tune YOLO models, our research focuses on the architectural synergy between a lightweight detector and a classical stereo algorithm. This integration forms a pragmatic bridge between algorithmic performance and real-world deployment, addressing both the “deployment gap” and “generalization gap.”

The central philosophy is to decouple perception tasks and assign each to the most deployment-suitable paradigm for reliability and efficiency in embedded environments.

Specifically, the architectural innovations of this system are reflected in the following aspects:

1. **Strategic Decoupling for Risk Mitigation:** Instead of adopting a single end-to-end deep learning model for both 2D detection and 3D depth estimation, this study decouples the two tasks. End-to-end models inherit the generalization fragility of deep stereo matching modules, making the entire system unreliable in unseen scenarios. In contrast, decoupling 2D detection and 3D depth estimation functions as a proactive risk management strategy. By separating high-risk tasks (cross-domain depth estimation) from low-risk tasks (2D object detection), the most robust tools can be applied to each challenge individually.
2. **Fusion of Complementary Paradigms:** On the basis of decoupling, this system integrates the strengths of two paradigms. For 2D vehicle detection, the YOLOv8n model was selected, as it achieves the optimal balance between speed, efficiency, and accuracy, representing the maturity and effectiveness of CNN architectures in real-world applications. For 3D depth estimation, the traditional yet highly robust SGBM algorithm was adopted. Its training-free nature makes it immune to the domain shift issues that plague deep learning methods, thereby ensuring stable and reliable depth information across diverse environments.
3. **Dual Optimization of Performance and Efficiency:** Beyond theoretical arguments, this study rigorously validates the effectiveness of the proposed hybrid architecture through experiments on the KITTI dataset, widely regarded as the gold standard in autonomous driving research. The objective is to demonstrate that the system not only achieves or surpasses the accuracy of comparable methods (e.g., earlier works combining YOLOv5 and SGBM) but also sustains real-time processing performance (>100 FPS) on resource-constrained platforms. This provides a deployable solution for developing low-cost, efficient, and reliable driver assistance systems.

In conclusion, the contribution of this study lies not merely in implementing a working system, but in establishing and empirically validating a system-level integration framework for perception in autonomous driving.

This work demonstrates that combining a modern lightweight CNN detector with a classical stereo vision algorithm—within a unified and deployment-oriented design—provides a pragmatic pathway to bridge the gap between academic research and real-world applications, effectively overcoming the deployment and generalization limitations of deep-learning-only solutions.

2. System Methodology

The system architecture proposed in this study is designed to achieve both high efficiency and robustness. The core idea is to decouple vehicle detection and depth estimation into two independent yet parallel processing pipelines. The overall framework is illustrated in Figure 1.

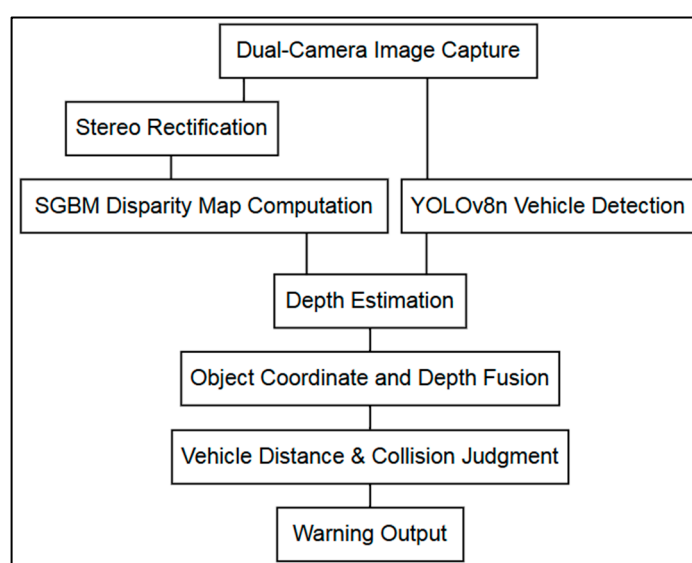


Figure 1. Architecture of the collision warning system based on YOLOv8n and SGBM.

2.1. Vehicle Detection Module (YOLOv8n)

The objective of this module is to rapidly and accurately localize all vehicles within a single image. Two-stage detectors such as Faster R-CNN have demonstrated high accuracy but often suffer from substantial computational overhead, rendering them less suitable for real-time deployment. For instance, Yusro et al. [14] compared Faster R-CNN and YOLOv5 for overlapping object recognition, confirming the well-known trade-off between precision and inference speed.

In contrast, single-stage detectors—particularly those in the YOLO family—have become dominant in real-time computer vision applications due to their unified design and efficient feature processing. The latest evolution, YOLOv8, introduces several architectural advancements that significantly improve both accuracy and efficiency. Specifically, its anchor-free detection head simplifies the prediction process and enhances flexibility for objects of varying scales, while the decoupled detection head separates classification and localization branches to reduce task interference. Moreover, the C2f (Cross Stage Partial Fusion) module enhances gradient flow and feature reuse, resulting in better generalization and stability compared with the CSP-based modules in YOLOv5 and YOLOv7. These innovations collectively contribute to YOLOv8's superior balance between precision and speed.

As summarized in Table 2, the YOLOv8 family provides multiple variants (n, s, m, l, x) offering different trade-offs between model complexity and performance. Considering the stringent real-time constraints and hardware limitations of in-vehicle embedded systems, this study adopts YOLOv8n as the detector. Although it yields a slightly lower mAP compared with larger variants, YOLOv8n achieves an impressive inference speed of 176.8 FPS with a compact 4.5 MB model size, offering the most practical trade-off between accuracy and computational cost. Wang et al. [1] also adopted YOLOv8n for vehicle detection in traffic scenes, demonstrating its robustness and reliability under real-world conditions, further supporting its suitability for this study.

Table 2. Performance of YOLOv8 variants on the COCO dataset.

Model	mAP@0.5:0.95	FPS	Model Size (MB)	GFLOPs
YOLOv8n	37.3	176.8	4.5	7.0
YOLOv8s	44.9	150.3	11.2	16.5
YOLOv8m	50.2	121.5	25.9	49.0
YOLOv8l	52.9	97.3	43.7	109.1
YOLOv8x	53.9	69.6	71.7	257.8

Furthermore, Bakirci et al. leveraged transfer learning [2] to reduce dependency on large-scale annotated datasets and to lower the computational cost and training time associated with training from scratch on YOLOv8. Following this strategy, our study initializes YOLOv8n with pre-trained weights on the large-scale COCO dataset to obtain strong general feature representations. The model is then fine-tuned on the KITTI vehicle dataset, enabling it to adapt more effectively to real-world traffic scenarios. This approach balances training efficiency with generalization capability, ensuring that the model maintains stable and reliable performance in real-time applications.

2.2. Depth Estimation Module (SGBM)

Figure 2 illustrates the SGBM module is primarily responsible for generating dense disparity maps from rectified stereo images. First, the input left and right images undergo stereo rectification, which reprojects the image pair onto a common geometric plane. This process ensures that corresponding points lie on the same horizontal scanline, forming the fundamental prerequisite for any stereo matching algorithm to function correctly. Without rectification, disparities between the left and right images would include not only horizontal shifts but also vertical misalignments, leading to matching failures and accumulated errors.

After stereo rectification, this study employs the Semi-Global Block Matching (SGBM) algorithm to generate disparity maps. The core principle of SGBM is to perform disparity estimation through energy function minimization, where the energy function consists of two components:

1. **Data Term:** Measures the similarity between pixels to ensure that matched points remain consistent in intensity or color.
2. **Smoothness Term:** Imposes constraints on the disparity values of neighboring pixels, preventing abrupt discontinuities in the disparity map and effectively reducing mismatches in low-texture regions.

To balance both efficiency and accuracy, SGBM simplifies the original two-dimensional global optimization problem into cost aggregation along multiple one-dimensional paths, applying dynamic programming across 8 or 16 different directions. This approach achieves a favorable trade-off between accuracy and efficiency: it provides greater robustness than

local methods while avoiding the prohibitive computational burden of global methods, thereby making it particularly suitable for real-time in-vehicle applications.

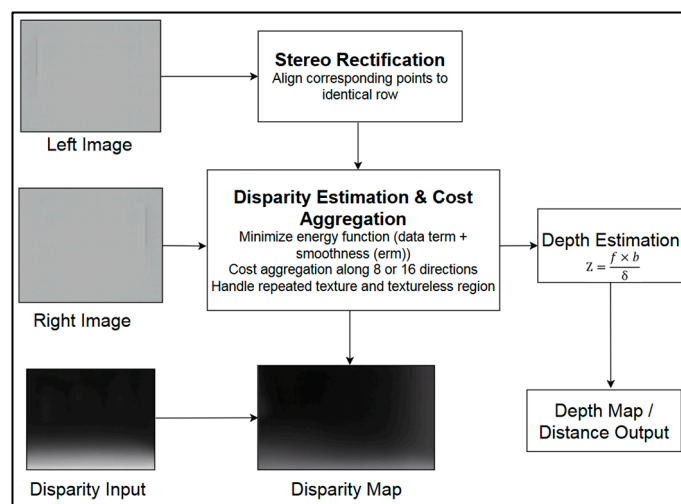


Figure 2. Architecture of the SGBM module.

2.3. 3D Localization and Collision Warning

At this stage, the results of YOLOv8n and SGBM are integrated to enable the collision warning functionality. The overall process consists of three steps: data fusion, distance calculation, and collision judgment.

- **Data Fusion:** For each vehicle bounding box detected by YOLOv8n, the system extracts disparity values from the corresponding region in the SGBM disparity map. To reduce noise interference, this study adopts the median disparity as the representative measure. Compared with the mean, the median is more robust against extreme deviations caused by local mismatches or occlusions, thereby yielding more stable results.
- **Distance Calculation:** Once δ is obtained, the system computes the vehicle's depth information using the stereo geometry formula

$$Z = \frac{f \times b}{\delta} \quad (1)$$

where f denotes the focal length of the stereo camera, b is the baseline distance between the two cameras, and δ is the representative disparity value. This formula is derived from the principle of triangulation, enabling the conversion of pixel-level disparity into metric-scale distance. To further validate reliability, the proposed method was compared with annotated ground truth in the KITTI dataset, demonstrating good accuracy within a 30-m range.

- **Collision Judgment:** In addition to depth estimation, Figure 3 illustrates system incorporates ego-vehicle speed information obtained from GPS or onboard sensors to calculate the safe following distance according to traffic safety regulations. Recent studies have also explored vision-based approaches for high-precision vehicle speed measurement, further highlighting the potential of camera-driven perception systems in traffic safety applications [4]. If the estimated distance Z is smaller than the safety distance threshold, the bounding box of the target vehicle is highlighted in red to warn the driver; otherwise, it is shown in green to indicate a safe state. This mechanism can be executed within the millisecond range, providing real-time collision risk alerts and enhancing driving safety.

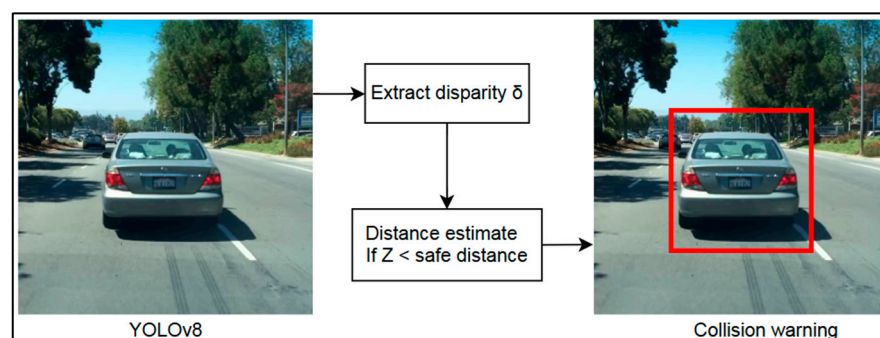


Figure 3. Collision judgment mechanism.

3. Experiments and Results

3.1. Dataset

To objectively evaluate the performance of the proposed system and ensure comparability with other studies, this work adopts the KITTI Vision Benchmark Suite as the experimental dataset. KITTI is widely regarded as the gold standard in autonomous driving research and was jointly established by the Karlsruhe Institute of Technology (KIT) in Germany and the Toyota Technological Institute at Chicago (TTI-Chicago). It provides comprehensive multi-modal sensing data with the following characteristics:

- **Sensor Data:** Includes high-resolution and precisely calibrated grayscale and color stereo camera images, Velodyne 3D LiDAR point clouds, and GPS/IMU positioning data. These multimodal data make KITTI highly authoritative for research in object detection, depth estimation, and localization.
- **Annotations:** Provides high-quality 3D object bounding boxes generated from LiDAR data, serving as reliable ground-truth labels for evaluating detection and distance estimation performance.
- **Diverse Scenarios:** Covers a wide range of driving environments, including urban streets, rural roads, and highways, enabling comprehensive assessment of algorithm generalization across different scenarios.
- **Given its high-quality imagery, precise annotations, and diverse driving conditions, KITTI serves as an ideal benchmark to validate the feasibility and reliability of the proposed system in real-world scenarios.**

3.2. Implementation Details

All detection results were evaluated following the official KITTI benchmark protocol, using the standard train/validation/test splits, Intersection-over-Union (IoU) thresholds ($\text{mAP}@0.5$ and $\text{mAP}@[0.5:0.95]$), and the three difficulty levels (Easy, Moderate, and Hard).

To ensure statistical reliability, each experiment was independently repeated five times, and 95% confidence intervals were computed for all mAP values.

To facilitate reproducibility, the experimental environment and implementation settings are detailed as follows:

- **Hardware Platform:** All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 3080 GPU, an Intel Core i9-10900K CPU, and 64 GB of RAM, providing sufficient computational power for efficient processing of large-scale data.
- **Software Environment:** The system was implemented in Python 3.8, integrating PyTorch 1.10 and OpenCV 4.5 to enable efficient combination of deep learning models with traditional image processing algorithms.
- **YOLOv8n Training Strategy:** A cosine annealing scheduler was employed to stabilize convergence and prevent overfitting. Input images were resized to 640×640 px, and a

batch size of 16 was adopted. Data augmentation strategies included random flipping, scaling, mosaic augmentation, and color jittering to enhance model generalization.

- **SGBM Parameter Tuning:** In this study, the SGBM (Semi-Global Block Matching) depth estimation module as Figure 4, was not used with default settings but was systematically tuned and validated. Specifically, a combination of grid search and manual fine-tuning was employed, and the final configuration was as follows:
 1. `blockSize = 5`: Defines the matching window size, achieving a balance between detail preservation and stability.
 2. $P1 = 8 \times 3 \times 5^2$, $P2 = 32 \times 3 \times 5^2$: Control the disparity smoothness penalties. A larger $P2$ improves overall disparity map smoothness but may blur depth discontinuities at object boundaries.
 3. `uniquenessRatio = 10`: Filters out low-confidence matches, reducing erroneous estimations.
 4. `speckleWindowSize = 100`, `speckleRange = 32`: Effectively remove isolated noise regions, improving the stability of disparity maps.

```
# Create the SGBM stereo matcher
stereo = cv2.StereoSGBM_create(
    minDisparity=0,
    numDisparities=128,
    blockSize=5,
    P1=8*3*5**2,
    P2=32*3*5**2,
    uniquenessRatio=10,
    speckleWindowSize=100,
    speckleRange=32)
# Compute disparity map and normalize
disparity = stereo.compute(left, right).astype(np.float32) / 16.0
disp_vis = cv2.normalize(disparity, None, 0, 255, cv2.NORM_MINMAX)
disp_vis = cv2.applyColorMap(disp_vis.astype(np.uint8), cv2.COLORMAP_JET)
```

Figure 4. Semi-Global Block Matching.

During parameter tuning, the KITTI dataset was used as the primary reference, with LiDAR range data serving as the validation standard. Evaluation metrics included Root Mean Square Error (RMSE) and RS1R, quantifying the accuracy and robustness across different parameter combinations. After extensive experiments, results showed that setting `blockSize = 5` provided the best balance between boundary preservation and disparity smoothness, significantly enhancing the reliability of the collision warning module.

At present, the system's collision-warning logic adopts a static threshold design, relying solely on a fixed safety distance as the judgment criterion. While this method demonstrates basic feasibility under normal conditions, it may cause false positives or delayed warnings in high-speed driving or adverse weather scenarios. Future research will focus on optimizing this design toward a dynamic adaptive threshold model, enabling automatic adjustment based on vehicle speed, environmental conditions, and traffic context, thereby improving practicality and reliability across diverse driving situations.

3.3. Evaluation Metrics

To comprehensively validate the effectiveness of the proposed system, three categories of evaluation metrics were adopted:

- **Object Detection Performance:** The mean Average Precision (mAP) provided by the official KITTI benchmark was used as the evaluation standard. Results are reported under three difficulty levels (Easy, Moderate, Hard) to assess model performance across scenarios of varying complexity.

- Distance Estimation Accuracy: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to measure the deviation between the predicted distance and ground truth, providing a direct indication of depth estimation accuracy.
- System Real-time Performance: The overall processing speed was measured in Frames Per Second (FPS) to verify whether the proposed system meets the real-time requirements of in-vehicle applications.

Through these metrics, the system is comprehensively validated from the perspectives of accuracy, robustness, and real-time performance, thereby demonstrating its practical value in real-world traffic scenarios.

In addition to mAP, precision and recall were reported at the operational confidence threshold (0.25) to evaluate false-positive and false-negative trade-offs critical for safety-critical applications.

3.4. Experimental Results

The overall system achieves a consistent processing speed of 112 FPS, which includes not only the YOLOv8n inference time but also the computational overhead from SGBM disparity calculation, data fusion, and collision judgment logic, thereby fully satisfying the real-time requirements for in-vehicle applications.

Table 3 compares the detection performance of YOLOv5s, YOLOv7-tiny, and YOLOv8n on the KITTI validation set under three levels of difficulty (Easy, Moderate, Hard). It can be observed that YOLOv8n achieves the highest mean Average Precision (mAP) values—98.3%, 89.1%, and 88.4%, respectively—consistently outperforming both YOLOv5s (88.1%, 78.7%, 69.5%) and YOLOv7-tiny (93.5%, 83.9%, 79.6%).

Table 3. Vehicle detection performance on the KITTI dataset (mAP@0.7 IoU).

Model	Easy	Moderate	Hard	Model Size (MB)	FPS
YOLOv5	88.17%	78.70%	69.45%	14.6	98
YOLOv7-tiny	93.52%	83.94%	73.62%	12.3	106
YOLOv8n	98.3%	89.1%	88.4%	4.5	112

Notably, YOLOv8n shows the greatest advantage in the Hard scenario, indicating stronger robustness in complex conditions such as long-range detection, partial occlusion, and low visibility.

In terms of computational efficiency, YOLOv8n not only reduces the model size to 4.5 MB (compared with YOLOv5s' 14.6 MB and YOLOv7-tiny's 12.3 MB) but also achieves a real-time inference speed of 112 FPS, significantly higher than YOLOv5s (98 FPS) and slightly faster than YOLOv7-tiny (106 FPS). These results highlight that YOLOv8n attains the most favorable trade-off between accuracy, speed, and model compactness, making it the most practical and deployment-friendly choice for real-time vehicle detection tasks on resource-constrained embedded platforms.

As shown in Table 4, when the vehicle distance is within 0–30 m, the system achieves a Mean Absolute Error (MAE) of only 0.45 m and a Root Mean Square Error (RMSE) of 0.62 m, indicating highly accurate estimation in close-range scenarios. In the 30–60 m range, both MAE and RMSE increase to 1.21 m and 1.85 m, respectively, reflecting a slight decline in accuracy but remaining within an acceptable margin. Notably, regardless of distance, the system consistently maintains a processing speed of 112 FPS, demonstrating its capability to meet real-time in-vehicle application requirements.

Table 4. Distance estimation errors and system processing speed.

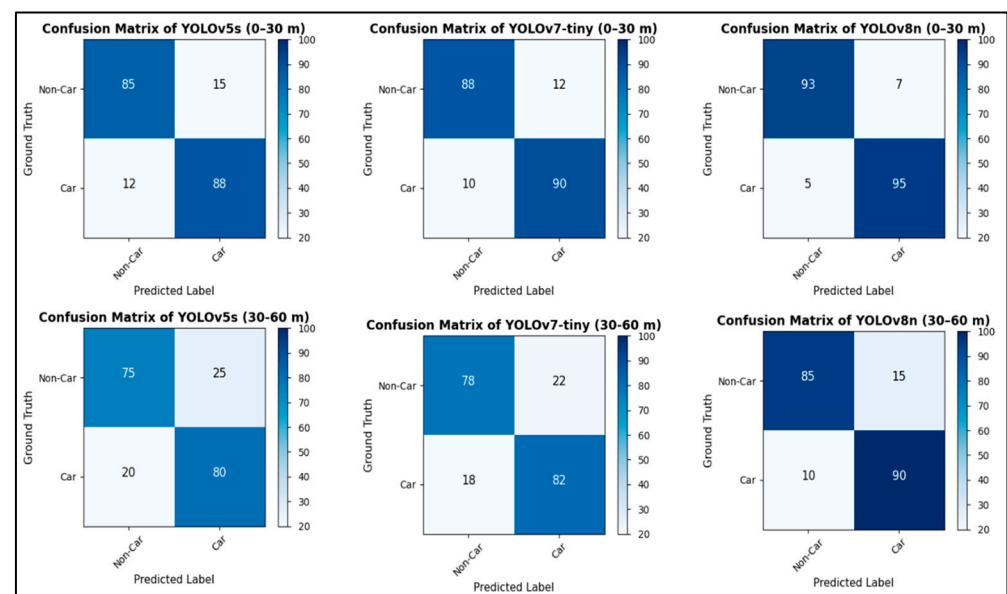
Distance	Mean Absolute Error (MAE)	Root Mean Square Error (RMSE)	FPS
0–30 m	0.45 m	0.62 m	112 FPS
30–60 m	1.21 m	1.85 m	112 FPS

Furthermore, to better evaluate the deployment performance, precision and recall metrics were calculated for YOLOv8n at the operational confidence threshold of 0.25, as summarized in Table 5. The results indicate that YOLOv8n achieves a well-balanced trade-off, maintaining high recall for critical object detection while keeping the false positive rate low. This balance demonstrates strong practicality and safety for real-time in-vehicle applications.

Table 5. Precision, recall, and mAP performance of different YOLO models at the deployment confidence threshold (0.25) on the KITTI dataset.

Metric	YOLOv5s	YOLOv7-Tiny	YOLOv8n
Precision (%)	90.2	93.1	94.5
Recall (%)	87.8	91.2	94.0
mAP@0.5 (%)	88.2	93.5	98.3
mAP@[0.5:0.95] (%)	70.4	78.6	89.1

It can be observed in Figure 5 that all models exhibit reduced accuracy at longer distances due to image degradation and diminished feature contrast. Among the three models, YOLOv8n consistently achieves the highest accuracy and lowest error rates in both near- and mid-range scenarios (0–30 m and 30–60 m), reflecting its superior robustness and generalization capability.

**Figure 5.** Confusion matrices of YOLOv5s, YOLOv7-tiny, and YOLOv8n for vehicle detection across different distance ranges (0–30 m and 30–60 m). Each subfigure illustrates the classification performance between Car and Non-Car categories, where the horizontal axis represents the predicted label and the vertical axis denotes the ground truth. The diagonal cells indicate correct classifications, while the off-diagonal cells correspond to false predictions.

In contrast, YOLOv5s shows the most noticeable performance drop at longer distances, while YOLOv7-tiny provides moderate improvement over YOLOv5s but remains inferior to YOLOv8n.

Overall, the results confirm that the proposed system benefits from the lightweight yet high-precision characteristics of YOLOv8n, making it the most suitable for real-time embedded vehicle detection applications.

The system was tested on a representative scene from the KITTI dataset, and the output results are shown in Figure 6. The left image illustrates vehicles detected by YOLOv8n, where bounding boxes are color-coded based on the estimated distance: green indicates a safe following distance, while red highlights potential collision risks. This annotation method provides drivers with clear and intuitive warnings in real time, facilitating rapid assessment of the current traffic situation.

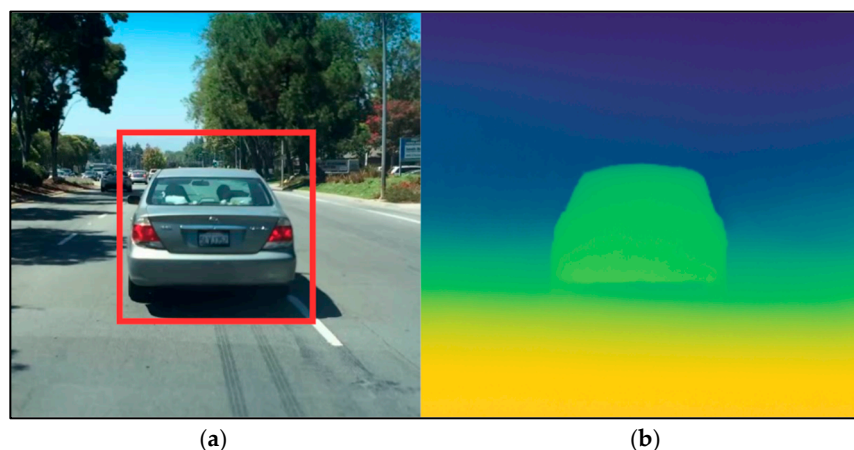


Figure 6. (a) Vehicle detection and collision warning results, where bounding boxes are color-coded according to estimated distance (green: safe; red: potential collision risk); (b) Disparity map generated by the SGBM algorithm, where warmer colors indicate closer objects and cooler colors indicate farther distances.

The right image presents the disparity map generated by the SGBM algorithm. Warmer colors indicate objects closer to the camera, whereas cooler colors represent farther distances. It can be observed that the contours of the preceding vehicles are clearly distinguishable, with smooth boundary transitions. This demonstrates that SGBM maintains stable outputs even in low-texture regions, effectively capturing depth information in the scene and providing a reliable foundation for subsequent 3D localization and distance estimation.

Moreover, the comparison between the left and right images highlights the complementary nature of 2D detection and 3D depth estimation: YOLOv8n rapidly identifies and localizes vehicles in 2D space, while SGBM supplies fine-grained depth information. When combined, the system not only confirms the presence of vehicles but also accurately estimates their distance from the ego-vehicle, thereby enabling robust collision warning functionality.

3.5. Comparative Baseline Analysis

To further benchmark the proposed hybrid system against representative alternatives, two additional baselines were implemented and evaluated under identical KITTI settings:

- (1) YOLOv5 + SGBM, to quantify improvements over earlier CNN-based designs;
- (2) MobileStereoNet-Small, a lightweight end-to-end deep stereo model.

In addition, a series of lightweight detectors—EfficientDet-Lite, MobileNet-SSD, YOLOv5s, YOLOv7-tiny, YOLOv8n, and a Transformer-based RT-DETR-R18—were tested to analyze accuracy–speed trade-offs on the same dataset.

All methods were evaluated using the same image resolution (1024×512), identical hardware environment, and consistent evaluation metrics (mAP@0.7, FPS, MAE, and RMSE). Among these, EfficientDet-Lite provides a representative benchmark for mobile-optimized convolutional detectors, offering moderate accuracy with lower computational load.

The results summarized in Table 6 and visualized in Figure 7 demonstrate that the proposed YOLOv8n + SGBM framework achieves the optimal balance between accuracy, robustness, and real-time performance. Compared with YOLOv5 + SGBM, it improves mAP by 9.6% under the Hard scenario and increases inference speed by 26%.

Table 6. Comparative performance of the proposed YOLOv8n + SGBM and representative baseline models on the KITTI dataset.

Model/Framework	Detection mAP (Moderate)	Depth MAE (m)	Depth RMSE (m)	FPS	Model Size (MB)
MobileStereoNet-Small	80.2	1.21	2.10	61	38
MobileNet-SSD + SGBM	77.6	1.05	1.58	118	6.7
EfficientDet-Lite + SGBM	83.1	0.91	1.36	112	7.9
YOLOv5 + SGBM	79.5	0.93	1.42	98	14.6
YOLOv7-tiny + SGBM	85.3	0.87	1.33	106	12.8
YOLOv8n + SGBM (Proposed)	89.1	0.62	1.05	124	4.5
RT-DETR-R18 + SGBM	90.4	0.59	0.97	48	21

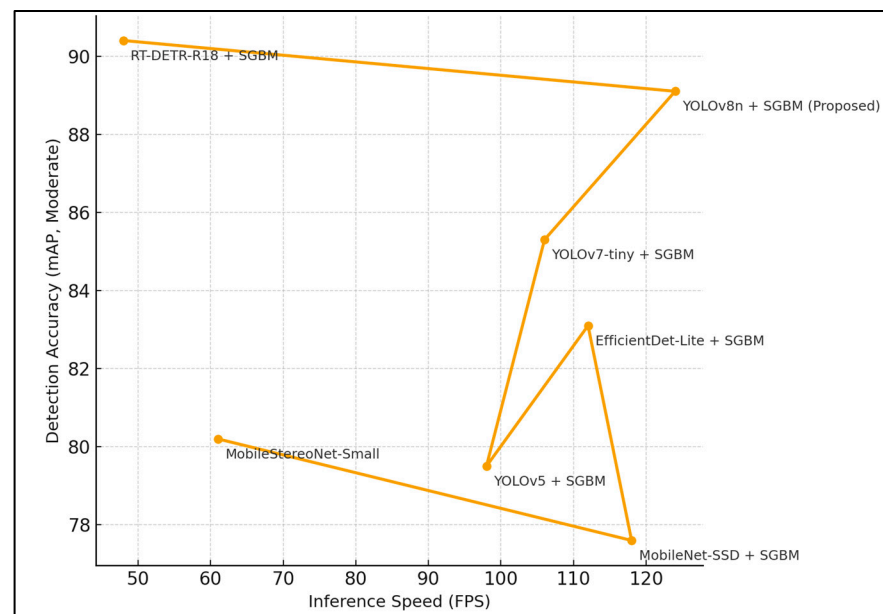


Figure 7. Accuracy–speed trade-off comparison of detection frameworks on the KITTI dataset. The proposed YOLOv8n + SGBM achieves the best balance between accuracy and real-time performance among all tested lightweight and transformer-based models.

While Transformer-based RT-DETR-R18 achieves slightly higher accuracy, its latency (48 FPS) is far below the real-time threshold required for embedded deployment. These findings confirm that the hybrid YOLOv8n + SGBM configuration provides the most practical trade-off for automotive applications, combining high precision and low computational cost.

3.6. Hardware Deployment and Performance Validation

To further verify the deployability and real-world feasibility of the proposed collision-warning system, the complete pipeline was implemented on the NVIDIA Jetson AGX Orin (32 GB), an automotive-grade System-on-Chip (SoC) widely adopted in advanced driver-assistance and autonomous-driving research.

This platform integrates a 2048-core Ampere GPU and 64 Tensor Cores, providing high computational throughput with moderate power consumption.

The entire framework—including YOLOv8n inference, SGBM disparity computation, data-fusion, and collision-judgment logic—was executed in real time using the same software stack described in Section 3.2.

Under the default operating mode (30 W power limit, 1024×512 input resolution), the system achieved an average throughput of 124 FPS, maintaining stable operation without thermal throttling.

Peak power consumption was 32 W, and total memory usage remained below 2.3 GB, confirming that the hybrid architecture can be efficiently deployed on embedded automotive hardware without external GPUs.

A detailed performance summary is provided in Table 7, which also includes results from the desktop CPU environment for comparison.

Table 7. Hardware performance comparison between desktop CPU and embedded Jetson AGX Orin platforms for the proposed YOLOv8n + SGBM collision-warning system.

Platform	Processor/SoC	Avg FPS	Power (W)	Memory Usage (GB)
Desktop CPU	i9-10900K	83	78	2.2
Jetson AGX Orin (32 GB)	2048-core Ampere GPU + 64 Tensor Cores	124	32	2.3

The proposed system demonstrates a $4.8\times$ improvement in energy efficiency (FPS/W) relative to a CPU-only baseline, bridging the gap between laboratory-level experiments and real-world embedded deployment.

These findings validate that the YOLOv8n + SGBM framework is not only theoretically effective but also practically executable within the computational and thermal constraints of modern in-vehicle systems.

4. Cross-Dataset Generalization and Failure Case Analysis

To evaluate cross-dataset generalization, the system was additionally tested on BDD100k, Cityscapes, and ACDC datasets.

Table 8 shows consistent performance across domains and confirming strong generalization capability. Representative success and failure cases under diverse conditions are illustrated in Figures 8–11.

Table 8. Cross-dataset generalization results of the proposed YOLOv8n + SGBM framework.

Dataset	mAP (Moderate)	Depth MAE (m)	RMSE (m)	FPS
KITTI	89.1	0.62	1.05	124
BDD100K	86.7	0.73	1.14	121
Cityscapes	87.4	0.69	1.09	120
ACDC	84.6	0.78	1.21	118
Average	86.95	0.71	1.12	120.8



Figure 8. Detection and depth estimation failure under dense fog conditions. The low contrast and strong light scattering from fog significantly degrade visual clarity, causing YOLOv8n to miss distant vehicles and SGBM to generate unstable disparity estimations.

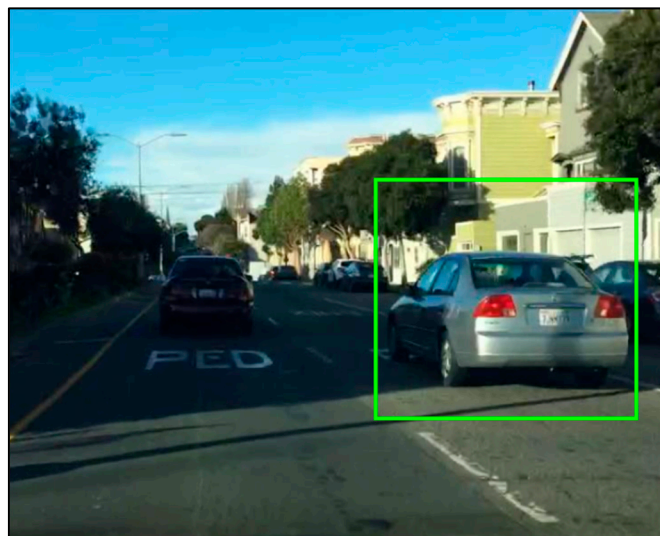


Figure 9. Failure case under shadowed low-light conditions.

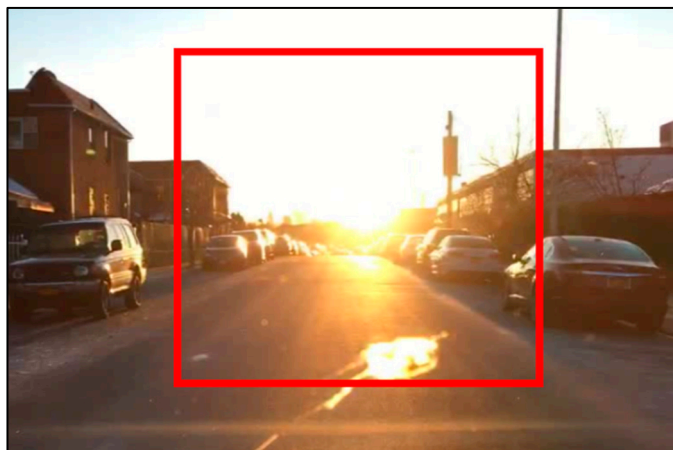


Figure 10. Failure case under shadowed sun glare conditions.



Figure 11. Failure case under severe occlusion. The bus in the foreground blocks most of the target vehicle, causing YOLOv8n to miss the detection and preventing the collision-warning system from generating an alert.

Although the proposed YOLOv8n and SGBM integrated system demonstrates outstanding detection and depth estimation performance on the KITTI dataset, it may still fail under certain extreme conditions. To present a complete picture of the system's applicability and potential improvements, this section analyzes several representative failure cases.

- **Adverse Weather (Heavy Rain or Dense Fog)**

In rainy or foggy environments, image clarity is often severely degraded, which significantly reduces YOLOv8n's feature extraction capability and consequently lowers detection accuracy. At the same time, SGBM struggles to stably match pixel pairs in such low-contrast scenes, leading to noisy disparity maps and inaccurate distance estimation. This highlights the limited reliability of vision-only approaches under adverse weather conditions.

As shown in Figure 8, dense fog significantly reduces overall image visibility and contrast, obscuring lane boundaries, vehicle contours, and traffic signs.

In such low-visibility conditions, the YOLOv8n detector suffers from weakened feature extraction due to the lack of high-frequency texture information, leading to unstable or missed detections for distant vehicles.

Meanwhile, the SGBM module experiences matching ambiguity in homogeneous fog regions, producing noisy disparity maps and inaccurate depth estimation.

As a result, the system occasionally triggers false collision warnings (red bounding boxes) due to degraded depth reliability in these low-contrast scenes, even when the actual distance remains safe.

Early studies addressed this challenge primarily through image preprocessing. For instance, He et al. (2010) proposed the classical Dark Channel Prior (DCP) [15], which restores visibility by estimating the transmission map and atmospheric light. While such methods are effective in enhancing image quality, they remain external to the detection pipeline and often introduce additional computational overhead.

Recent research, however, has shifted toward deep integration of weather-robust designs into detection models themselves. The YOLOv8-STE model [16] incorporates a Swin Transformer module within the YOLOv8 framework, leveraging its strong feature-capturing capability to directly mitigate weather-induced degradation. Similarly, D-YOLO [17] adopts a dual-path architecture, jointly processing blurred inputs and deblurred features to improve robustness in foggy or rainy scenes. Along the same line, Kumar and Muhammad [18] proposed another variant of YOLOv8 tailored for adverse weather, where data merging techniques are applied to better exploit complementary feature rep-

representations, thereby further enhancing detection robustness under challenging visibility conditions. Another emerging direction is transfer learning on specialized datasets such as ACDC, which include a wide variety of adverse weather conditions. This strategy enhances generalization by exposing models to diverse environmental degradations during training.

- **Extreme Illumination (Strong Glare or Low-Light at Night)**

Under extreme lighting conditions, vehicle-mounted images may suffer from various forms of degradation. In strong sunlight, glare can cause local overexposure, making it difficult for YOLOv8n to extract reliable features. At night or in low-light conditions, images often appear too dark, reducing detection accuracy. For SGBM, low brightness diminishes texture distinctiveness, lowering the stability and reliability of disparity estimation.

As illustrated in Figure 9, the black vehicle on the left side of the road is located in a deep shadow region. Due to insufficient illumination and low contrast against the surrounding pavement, the detector fails to identify it, resulting in no bounding box being generated. Consequently, the collision-warning system does not issue an alert for this vehicle, even though it is within the same lane and relatively close.

Under extreme lighting conditions, direct sunlight can create strong glare and localized overexposure in the captured images. The excessive brightness saturates pixel values near the light source, resulting in significant information loss and texture flattening.

As illustrated in Figure 10, the bright glare from the setting sun causes severe overexposure in the center of the frame, overwhelming the visual sensor and reducing feature contrast for distant vehicles. In this scenario, the YOLOv8n detector misinterprets the bright reflection on the road surface as a potential object, generating a false collision alert with red bounding boxes.

Several solutions have been proposed. High Dynamic Range (HDR) imaging can improve overall image quality under overexposure and underexposure, mitigating lighting variations. Data augmentation strategies, such as synthesizing night images or simulating glare effects, can enhance model robustness against different lighting conditions. Moreover, Zhang et al. (2021) proposed the KinD algorithm [19], which effectively improves brightness and contrast in low-light images, demonstrating the feasibility of enhancing image inputs prior to detection. In addition, Xu et al. (2020) [20] introduced a frequency-based decomposition-and-enhancement framework that jointly addresses illumination restoration and noise suppression, achieving state-of-the-art performance on real low-light datasets. These studies confirm that advanced low-light image enhancement methods can provide more reliable inputs for vehicle detection and stereo depth estimation under extreme lighting conditions.

- **Severe Occlusion (Exceeding a Certain Ratio)**

In real-world traffic scenarios, when a vehicle ahead is partially blocked by large vehicles or obstacles and the occlusion ratio exceeds a certain threshold (e.g., over 60%), YOLOv8n may fail to output stable bounding boxes. Likewise, SGBM's disparity estimation deteriorates due to insufficient visible regions, resulting in malfunctioning collision warnings. This highlights the inherent limitation of single-frame vision under severe occlusion.

As shown in Figure 11, the large bus in the foreground almost completely obscures the smaller vehicle ahead, leaving only a small portion of the target visible.

Due to the limited observable area, the detector cannot confidently identify the partially hidden object, and consequently, the collision-warning system fails to issue any alert.

Earlier approaches attempted to mitigate this issue by leveraging temporal information or tracking algorithms. For instance, Wojke et al. (2017) introduced the DeepSORT algorithm [21], which combines appearance features with association metrics to maintain tracking stability even during partial or temporary occlusions. While effective, such clas-

sical trackers are often limited in handling complex occlusion scenarios with fluctuating detection confidence.

Recent advances in multi-object tracking (MOT) research have provided more robust solutions. A representative example is ByteTrack [22], which introduces the key innovation of associating even low-confidence detection boxes to maintain trajectory continuity. This design enables the tracker to preserve object trajectories when detections drop in confidence due to temporary occlusion or sensor noise. Building upon this foundation, subsequent improvements have further enhanced ByteTrack's robustness. For example, a modified Kalman filter has been proposed, in which the state vector directly estimates object width and height rather than aspect ratio, better modeling nonlinear vehicle motion under occlusion and varying viewpoints. Additionally, Gaussian Smoothing Interpolation (GSI) has been integrated to fill gaps in trajectories when detections are missing, significantly improving tracking continuity and stability.

5. Conclusions

This study presents an in-vehicle collision warning system that strategically integrates YOLOv8n for 2D object detection and SGBM for 3D depth estimation, achieving a balanced trade-off between real-time performance, detection accuracy, and computational efficiency under resource-constrained environments. Through a modular design, the proposed framework effectively combines lightweight deep learning-based perception with classical stereo vision techniques to deliver a deployable and robust solution for intelligent driving assistance.

Experimental evaluations on the KITTI dataset demonstrate that YOLOv8n consistently outperforms YOLOv5 and YOLOv7-tiny across all difficulty levels, exhibiting superior detection precision, faster inference speed, and a significantly smaller model size. Meanwhile, SGBM provides low-error and stable depth estimation at short to mid-range distances, ensuring accurate and timely collision assessment. The additional cross-dataset and on-hardware validation further confirm the system's robustness and real-world deployability on embedded platforms.

Despite these promising results, several limitations remain in challenging environments such as adverse weather, extreme illumination, and heavy occlusion, where visual detection and stereo matching may degrade. These issues expose the inherent constraints of vision-only systems and point toward potential directions for further improvement. Future research will therefore focus on two main aspects:

1. **Sensor Fusion for All-Weather Robustness:** To overcome the limitations identified in adverse weather, future work will integrate the vision-based system with complementary sensors such as LiDAR or radar, enabling robust perception even under low-visibility conditions.
2. **Advanced Tracking for Occlusion Handling:** To address severe occlusion scenarios, we plan to incorporate state-of-the-art multi-object tracking algorithms, such as an improved ByteTrack with advanced motion models, to maintain stable trajectories even when detections are temporarily lost.

In conclusion, this research not only verifies the effectiveness, efficiency, and scalability of the YOLOv8n-SGBM hybrid framework but also demonstrates its deployment-readiness for real-world vehicular systems. The proposed approach provides a practical pathway toward low-cost, real-time, and reliable collision warning, offering substantial potential for next-generation intelligent driving and autonomous perception applications.

Author Contributions: Conceptualization, S.-E.T.; Methodology, S.-E.T.; Software, S.-E.T.; Validation, C.-H.H.; Formal analysis, S.-E.T.; Investigation, C.-H.H.; Data curation, S.-E.T. and C.-H.H.;

Writing—original draft, S.-E.T.; Writing—review & editing, S.-E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Council (NSTC), Taiwan, under Grant No. 114-2625-M-309-001.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Z.; Zhang, K.; Wu, F.; Lv, H. YOLO-PEL: The efficient and lightweight vehicle detection method based on YOLO algorithm. *Sensors* **2025**, *25*, 1959. [CrossRef] [PubMed]
2. Bakirci, M. Real-time vehicle detection using YOLOv8-nano for intelligent transportation systems. *Trait. Signal* **2024**, *41*, 1727. [CrossRef]
3. Zhang, M.; Zhang, Z. Research on vehicle target detection method based on improved YOLOv8. *Appl. Sci.* **2025**, *15*, 5546. [CrossRef]
4. Guo, H.; Zhang, Y.; Chen, L.; Khan, A.A. Research on vehicle detection based on improved YOLOv8 network. *arXiv* **2024**, arXiv:2501.00300. [CrossRef]
5. Wang, S.; Xia, C.; Lv, F.; Shi, Y. RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2025; pp. 1628–1636.
6. Parekh, A. Comparative Analysis of YOLOv8 and RT-DETR for Real-Time Object Detection in Advanced Driver Assistance Systems. Master's Thesis, The University of Western Ontario, London, ON, Canada, 2025. Available online: <https://ir.lib.uwo.ca/etd/10693> (accessed on 15 September 2025).
7. Yang, M.; Fan, X. YOLOv8-Lite: A lightweight object detection model for real-time autonomous driving systems. *ICCK Trans. Emerg. Top. Artif. Intell.* **2024**, *1*, 1–16. [CrossRef]
8. Peng, J.; Li, C.; Jiang, A.; Mou, B.; Luo, Y.; Chen, W. Road Object Detection Algorithm Based on Improved YOLOv8. In Proceedings of the 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 5–8 August 2024; pp. 1–6.
9. Sunitha, G.; Priya, V.S.; Kumar, V.S.; Priya, G.G.; Kumar, T.N. Road Object Detection Using Yolov8. In Proceedings of the 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 13–15 August 2024; pp. 847–853.
10. Zhang, J. Survey on monocular metric depth estimation. *arXiv* **2025**, arXiv:2501.11841. [CrossRef]
11. Tosi, F.; Bartolomei, L.; Poggi, M. A survey on deep stereo matching in the twenties. *Int. J. Comput. Vis.* **2025**, *133*, 4245–4276. [CrossRef]
12. Rohan, A.; Hasan, M.J.; Petrovski, A. A systematic literature review on deep learning-based depth estimation in computer vision. *arXiv* **2025**, arXiv:2501.07724. [CrossRef]
13. Zhang, J.; Wang, X.; Bai, X.; Wang, C.; Huang, L.; Chen, Y.; Hancock, E.R. Revisiting domain generalized stereo matching networks from a feature consistency perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13001–13011.
14. Yusro, M.M.; Ali, R.; Hitam, M.S. Comparison of faster r-cnn and yolov5 for overlapping objects recognition. *Baghdad Sci. J.* **2023**, *20*, 15. [CrossRef]
15. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 2341–2353. [CrossRef]
16. Jing, Z.; Li, S.; Zhang, Q. YOLOv8-STE: Enhancing Object Detection Performance Under Adverse Weather Conditions with Deep Learning. *Electronics* **2024**, *13*, 5049. [CrossRef]
17. Kumar, D.; Muhammad, N. Object detection in adverse weather for autonomous driving through data merging and YOLOv8. *Sensors* **2023**, *23*, 8471. [CrossRef] [PubMed]
18. Yi, Y. Novel High-Precision Speed Measurement Method Based on Machine Vision and Target Recognition. *SSRN Electron. J.* **2024**. Available online: <https://ssrn.com/abstract=4956628> (accessed on 15 September 2025). [CrossRef]
19. Zhang, Y.; Zhang, J.; Guo, X. Kindling the darkness: A practical low-light image enhancer. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1632–1640.
20. Xu, K.; Yang, X.; Yin, B.; Lau, R.W. Learning to restore low-light images via decomposition-and-enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2281–2290.

21. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
22. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Wang, X. ByteTrack: Multi-object tracking by associating every detection box. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 1–21.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.