

Article

Validating DVS Application in Autonomous Driving with Various AEB Scenarios in CARLA Simulator

Jingxiang Feng ¹, Peiran Zhao ², Jessada Konpang ^{1,*} , Adisorn Sirikham ^{1,*} , Haoran Zheng ^{3,4} , Phuri Kalnaowakul ⁵  and Jia Wang ⁶

¹ Faculty of Engineering, Rajamangala University of Technology Krungthep, Bangkok 10120, Thailand

² Department of Electrical, Computer, and Software Engineering, University of Auckland, Auckland 1010, New Zealand

³ Department of Chemical & Materials Engineering, University of Auckland, Auckland 1010, New Zealand

⁴ Department of Intelligent Vehicles, Guangzhou Panyu Polytechnic, Guangzhou 511483, China

⁵ Faculty of Industry and Technology, Rajamangala University of Technology Isan, Sakon Nakhon 47160, Thailand

⁶ School of Automobile Engineering, Foshan Polytechnic, Foshan 528000, China

* Correspondence: jessada.k@mail.rmutk.ac.th (J.K.); adisorn.s@mail.rmutk.ac.th (A.S.)

Abstract

Predicting potential collisions with leading vehicles is a fundamental capability of autonomous and assisted driving systems. In particular, automatic emergency braking (AEB) demands reaction times on the order of microseconds. A key limitation of existing approaches lies in their update rate, which is constrained by the sampling speed of conventional sensors. Event-based Dynamic Vision Sensors (DVSs), with their microsecond temporal resolution and high dynamic range, offer a promising alternative to frame-based cameras in challenging driving environments. In this work, we investigate the integration of DVS into autonomous driving pipelines, focusing specifically on AEB scenarios. Building on our earlier work, where a YOLO-based detection model was trained on real-world DVS data, we extend the approach to CARLA's simulated DVS environment. We publish a CARLA-compatible 2-channel DVS dataset aligned with our detection model, bridging the gap between real-world recordings and simulation. Through a series of simulated AEB scenarios, we demonstrate how DVS enables earlier and more reliable detection compared to RGB cameras, resulting in improved braking performance.

Keywords: event camera; neuromorphic vision; object detection; autonomous driving; detection-to-control; simulation



Academic Editor: Grzegorz Sierpiński

Received: 13 October 2025

Revised: 8 November 2025

Accepted: 14 November 2025

Published: 20 November 2025

Citation: Feng, J.; Zhao, P.; Konpang, J.; Sirikham, A.; Zheng, H.; Kalnaowakul, P.; Wang, J. Validating DVS Application in Autonomous Driving with Various AEB Scenarios in CARLA Simulator. *World Electr. Veh. J.* **2025**, *16*, 634. <https://doi.org/10.3390/wevj16110634>

Copyright: © 2025 by the authors. Published by MDPI on behalf of the World Electric Vehicle Association. 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/>).

1. Introduction

Conventional frame-based cameras, the backbone of most vision-based perception systems, struggle precisely under the conditions where safety matters most. In high dynamic range (HDR) environments—such as exiting a tunnel into bright sunlight or driving at night under low illumination—their limited dynamic response and fixed-rate sampling lead to sensor saturation, motion blur, and delayed perception. These weaknesses critically undermine safety functions in autonomous vehicles such as Automatic Emergency Braking (AEB), which depend on accurate, low-latency visual feedback to avoid collisions in high-speed or low-visibility scenarios.

Autonomous driving, a key frontier in intelligent transportation, promises to enhance both safety and efficiency on public roads [1]. Achieving this vision relies on a perception stack that fuses data from multiple sensor modalities, including cameras, LiDAR,

radar, and ultrasonic sensors [2]. Among these, vision sensors play a central role by providing dense semantic information essential for object detection, scene understanding, and decision making [3]. However, the inherent fragility of conventional cameras under HDR or fast-motion conditions raises concerns about the robustness of perception in safety-critical applications. [4,5]

Event-based Dynamic Vision Sensors (DVSs) have emerged as a compelling alternative to address these limitations. Unlike traditional cameras that capture full image frames at fixed intervals, DVSs operate asynchronously, recording only pixel-level brightness changes with microsecond temporal resolution [6]. This design offers several intrinsic advantages: resistance to motion blur, superior HDR performance, and reduced data redundancy. These characteristics make event-based cameras particularly well suited for augmenting or even replacing conventional RGB sensors in real-world autonomous driving, where rapid, reliable perception is vital for safety.

Building upon this motivation, we focus on Automatic Emergency Braking (AEB)—a canonical safety-critical function that directly depends on timely and accurate perception. AEB systems are typically designed around the concept of time-to-collision (TTC), the estimated time remaining before the ego vehicle collides with an object ahead. TTC can be computed directly if both the absolute distance and relative velocity of the leading object are known, which explains why LiDAR- and radar-based approaches have been widely adopted in commercial implementations [7,8]. While effective, these sensors primarily provide geometric rather than semantic information, leaving gaps in recognizing object type or predicting intent. Moreover, TTC-based systems remain vulnerable in edge cases where perception latency or sensor degradation delays detection—precisely the conditions where DVS can provide an advantage. In this work, we explore the potential of event-based vision for enhancing AEB performance under such challenging conditions. Our main contributions are:

- **Dataset Contribution:** We adapt our previously trained YOLO-based DVS detection model to the CARLA simulator and release a new two-channel CARLA-DVS dataset tailored for event-based perception. This dataset bridges the gap between real-world DVS benchmarks and simulation-based evaluation, enabling reproducible testing of DVS models in closed-loop driving scenarios.
- **Demonstration of DVS Advantages:** Through controlled experiments, we systematically analyze scenarios with high dynamic range, low illumination, and short reaction time, demonstrating that DVSs achieve earlier and more robust detection compared to conventional RGB cameras under these critical conditions.
- **Integrated Event-Based AEB System:** We present, to our knowledge, one of the first closed-loop AEB pipelines integrating event-based vision (DVS), YOLOv11 detection, and a conservative depth fusion strategy within the CARLA simulator. This integration enables both semantic awareness and geometric estimation under extreme conditions. The resulting system not only validates the feasibility of event-based AEB in simulation but also establishes a benchmark framework for future research on event-driven perception and control.

2. Related Work

2.1. Event Cameras in Autonomous Driving

The compelling advantages of event cameras have motivated a significant body of research aimed at harnessing this technology for automotive perception. The comprehensive survey by Gallego et al. [6] provides a foundational overview of event-based vision, systematically outlining the principles and algorithms developed for a wide range of computer vision tasks. The survey highlights how the unique properties of event cameras can

address critical challenges in autonomous driving, such as detecting fast-moving objects, robustly tracking features during aggressive maneuvers, and perceiving scenes with extreme lighting variations, like entering or exiting a tunnel. The work collates evidence showing that event data can be used to achieve state-of-the-art performance in feature tracking, ego-motion estimation, and dense 3D reconstruction, all of which are cornerstone tasks for vehicle localization and navigation. Building directly on this potential, recent work by Gehrig and Scaramuzza [9] demonstrates a practical and impactful application of these principles. They developed a low-latency, end-to-end automotive vision pipeline that leverages a recurrent neural network (RNN) to process event streams for object detection and tracking. Their key contribution is showing that this event-based system can perceive and react to new information with a latency reduction of up to an order of magnitude compared to state-of-the-art, frame-based systems. In challenging HDR and low-light scenarios, their pipeline successfully detected pedestrians and other vehicles 50 to 280 milliseconds faster than conventional camera systems. This significant reduction in reaction time is critical for safety-critical applications like AEB and collision avoidance, proving that event cameras can be a vital component in next-generation perception systems designed to overcome the inherent limitations of traditional sensors.

2.2. Event-Based Traffic Detection

A primary challenge in event-based vision is effectively modeling the asynchronous, sequential nature of the data. To address this, recurrent architectures have become prominent. The Recurrent Event-Camera Detector (RED) [10] stands out by processing raw event streams directly, bypassing the need for image reconstruction. By incorporating a temporal consistency loss, its recurrent design achieves high accuracy and outperforms traditional feed-forward methods. More recently, the Recurrent Vision Transformer (RVT) [11] has pushed performance further, achieving state-of-the-art results. RVT's efficient, multi-stage design provides a compelling combination of fast inference, low latency, and high-accuracy detection, demonstrating the power of transformer models in this domain. Another promising direction involves creating hybrid systems that combine the strengths of event cameras and traditional frame-based cameras. The Joint Detection Framework (JDF) [5] exemplifies this synergistic approach. It fuses the two modalities within a Convolutional Neural Network (CNN), using Spiking Neural Networks (SNNs) to generate visual attention maps. This novel technique enables effective synchronization between the sparse event stream and the dense image frames. The success of JDF underscores the significant benefits of integrating complementary sensor data, creating a more robust perception system for automotive applications. Similarly, another mixed-frame/event detector [12] utilizes a Dynamic and Active Pixel Sensor (DAVIS) to concurrently capture grayscale frames and events. By processing both streams with YOLO-based models and merging the outputs via a confidence map fusion method, the framework achieves higher accuracy and lower latency than systems relying solely on conventional cameras. These hybrid methods underscore the synergistic benefits of integrating complementary sensing modalities for more robust automotive perception.

Our work focuses on a CNN-based approach, which provides a practical and efficient solution for real-time autonomous driving applications. As established in our previous work [13], we preprocess raw event streams into dense, 2-channel event frame representations. This methodology allows us to leverage the high performance of well-established CNN architectures. We deliberately adopted a frame-based detection pipeline rather than asynchronous, event-native architectures such as RED or RVT for two main reasons. First, our entire “detection-to-control” framework (Section 3.6) is implemented within the CARLA simulator, which inherently operates on high-frequency event frames rather

than continuous asynchronous event streams. Consequently, a frame-based detector was a natural and necessary choice to ensure seamless integration with the rest of the system. Second, among frame-based detectors, YOLOv11 was selected because it offers an excellent trade-off between detection accuracy and inference latency, providing the real-time performance required for a time-critical AEB control loop. More importantly, the architecture's advanced Convolutional block with Parallel Spatial Attention (C2PSA) attention mechanism is particularly well-suited to our 2-channel event representation, as it emphasizes salient spatial regions corresponding to sparse "on" and "off" event clusters—precisely where the most informative motion cues reside.

2.3. Control from Event-Based Vision

The main limitation of the aforementioned studies is their focus on evaluating algorithmic performance using pre-recorded event data. While the offline approaches provide valuable benchmarks for accuracy and robustness, they largely overlook the practical challenges of real-time execution and system integration within a live autonomous driving stack. This absence of a system-level perspective limits the direct applicability of these methods in dynamic, real-world scenarios.

A significant and expanding area of research in vision-based robotics involves using event cameras to learn reactive policies, which directly convert sparse event data into control commands. This approach has unlocked new capabilities in highly dynamic environments. Fast, reactive policies have been successfully implemented in several challenging robotic tasks. For example, quadruped robots have been trained with reinforcement learning to catch fast-moving objects [14]. In aerial robotics, model-based algorithms have enabled drones to avoid fast, dynamic obstacles while hovering [15].

Instead of learning a direct end-to-end mapping from vision to control, many robust systems decompose the problem by using object detection as a critical intermediate task. This approach provides a structured, semantic understanding of the environment, which is more interpretable and can directly inform downstream planning and control modules. The importance of this strategy is evident in prior work on obstacle avoidance and navigation. For example, Zhang et al. [16] utilized an SNN with a depth camera for object detection to achieve the necessary scene awareness for safe maneuvering. Likewise, in the high-speed context of drone racing, Andersen et al. [17] trained a network to detect gates, as identifying the target's precise location is a prerequisite for generating a successful control policy.

Our previous work developed a YOLO-based detector and integrated this perception module into a CARLA-ROS simulation environment, demonstrating the feasibility of event-based detection within autonomous driving systems. This paper builds directly on that foundation, aiming to push beyond open-loop simulation to address the critical next steps: validating the perception pipeline in closed-loop scenarios and integrating its output directly with the vehicle's control system. By doing so, we demonstrate a complete, real-time perception-to-action framework, moving from algorithmic theory to practical application.

2.4. Depth Integration and AEB Systems

While event cameras have demonstrated impressive performance in high-speed reactive tasks—most notably in microsecond-level obstacle avoidance with drones—their integration into autonomous driving remains largely in the research domain. Nonetheless, the ability to capture fast dynamics with minimal latency highlights their potential for life-critical automotive functions such as AEB.

Conventional AEB systems primarily depend on radar and LiDAR, as these sensors provide robust range and velocity measurements necessary for accurate TTC estimation.

Safety assessments such as the Euro NCAP test protocols [18] further underscore the central role of precise distance and TTC calculations in certifying system reliability. The research on the TTC problem, or more generally speaking, obstacle avoidance using event-based vision has received large attention from the community of robotics [15,19] and neuromorphic engineering. In the field of autonomous driving, Sun et al. [20] introduce EvTTC, the first multi-sensor dataset for event-based time-to-collision estimation, offering an essential resource for developing and validating event-driven AEB algorithms.

TTC in autonomous driving could be obtained either from directly estimating event cameras or through the vision-depth fusion pipeline. Li et al. [21] introduce an event-based approach to time-to-collision estimation, using a coarse-to-fine geometric model fitting pipeline that efficiently handles the partial observability of event-based data. Their method outperforms conventional frame-based techniques in both accuracy and efficiency, highlighting the promise of neuromorphic sensors for safety-critical autonomous driving tasks. Beyond direct TTC estimation, another important direction is multi-sensor fusion pipelines that integrate detection and depth to achieve robust obstacle awareness. As reviewed by Wang et al. [22], camera–LiDAR–radar fusion exploits the complementary strengths of different modalities: cameras provide rich semantic information for object recognition, while LiDAR and radar contribute precise distance and velocity cues. Based on the level of data abstraction, fusion techniques are broadly categorized into feature-level and proposal-level fusion. The former aims to create a joint representation from raw or low-level data, while the latter focuses on collaborative decision-making using high-level reasoning. Feature-level fusion is generally more advantageous for tasks dependent on fine-grained information (e.g., pose estimation) where modal alignment is feasible. Conversely, proposal-level fusion is more suitable when modal differences are significant (e.g., text with images) or when fault tolerance is critical (e.g., sensor redundancy). It is summarized by Table 1 below.

Table 1. Comparison of Multi-Sensor Fusion Algorithms.

Category	Algorithm	Advantages	Limitations
Feature-Level Fusion	BEVFusion [23]	Provides intuitive spatial information, simplifying complex scene processing.	Highly sensitive to sensor calibration errors, which can degrade fusion quality.
	SimpleBEV [24]	Simple architecture enables easy implementation and deployment.	Performance may lag behind complex models in challenging scenes.
	FusionFormer [25]	Self-attention mechanism enhances global context modeling.	High computational overhead can hinder real-time performance.
Proposal-Level Fusion	F-PointNet [26]	Efficient and robust processing of point cloud regions of interest.	High computational complexity can be a bottleneck for real-time application.
	FSF [27]	Adaptive feature selection improves fusion effectiveness and flexibility.	Complex implementation; performance depends heavily on initial feature quality.
	SparseFusion [28]	Specializes in sparse data, reducing the overall computational load.	Potential for information loss in extremely sparse scenarios.

The work presented underscores that reliable AEB triggering depends not only on accurate ranging but also on the semantic filtering of safety-critical objects, motivating our integration of detection with depth cues for more robust time-to-collision prediction. At the same time, the comparison in the table indicates that improved fusion performance often

comes at the cost of higher computational demand—a trade-off that needs to be carefully tuned in real-time applications such as autonomous driving. This highlights the importance of lightweight design and real-time efficiency as core requirements for our system.

2.5. CARLA DVS Simulation

The event-based sensor model within the CARLA simulator is designed to emulate a DVS featuring a 140 dB dynamic range and a temporal resolution on the scale of microseconds. The sensor operates by asynchronously generating events, denoted as $e = (x, y, t, p)$, from individual pixels. An event is triggered at pixel coordinates (x, y) at a specific time t whenever the change in logarithmic intensity, L , surpasses a predefined contrast threshold, C . This triggering mechanism is mathematically expressed as:

$$L(x, y, t) - L(x, y, t - \delta t) = p \cdot C \quad (1)$$

Here, $t - \delta t$ is the timestamp of the most recent event at that pixel, and the polarity p indicates the direction of the brightness change—positive for an increase and negative for a decrease. For our experiments, this contrast threshold was set to $C = 0.3$. In visualizations, these events are typically represented by blue dots for positive polarity and red dots for negative, effectively mapping the dynamic elements of a scene over time.

CARLA's DVS implementation generates events by performing uniform temporal sampling between two consecutive synchronous ticks of the simulation world. This approach is conceptually aligned with our data preprocessing pipeline and means that the simulator internally handles the conversion of raw asynchronous data into structured event frames or tensors.

However, to faithfully replicate the microsecond-level temporal resolution of a physical event camera, the simulated sensor must operate at a much higher frequency than typical frame-based cameras. The rate of event generation is directly proportional to the dynamics within the scene; for instance, a faster-moving vehicle will produce a denser stream of events. Consequently, to preserve temporal accuracy, the sensor's operating frequency must be dynamically scaled with the motion intensity. This leads to a crucial trade-off between temporal fidelity and computational efficiency. While a higher frequency yields more precise data, it also imposes a greater processing burden. Conversely, a lower frequency reduces computational load at the risk of compromising the perceptual accuracy of the event stream.

A known limitation of simulated data is the “sim-to-real” gap. Research by Tan et al. [29] highlighted this issue, concluding that while CARLA's DVS is a valuable tool, its synthetic data cannot entirely replace real-world data for training robust object detection models. Building on this, our previous work involved a systematic analysis of this domain gap, focusing on the data's modal characteristics. Our investigation confirmed the existence of a domain gap but also uncovered compelling structural similarities between the simulated and real-world datasets. By comparing the feature distribution shapes and variance, we observed that both domains exhibit analogous scatter patterns within the principal component space, with a standard deviation ratio of approximately 1:1.14. This finding suggests that while the domains have distinct central positions, their internal feature structures are highly comparable. This underlying structural parallel provides a strong justification for our transfer learning strategy, as it confirms both the necessity and the feasibility of leveraging shared features across domains while adapting to their specific differences.

3. Methodology

3.1. YOLOv11-Based Event Detection

We extend our detector to CARLA's DVS environment by adapting the state-of-the-art YOLOv11 architecture. The input layer of the model is modified to accept a 2-channel format, which corresponds to the accumulated positive and negative events from the DVS. YOLOv11 introduces several architectural innovations that significantly improve detection accuracy while maintaining real-time performance.

The YOLOv11 architecture is composed of three primary components: a Backbone, a Neck, and a Head. For its backbone, YOLOv11 introduces a new C3k2 block, a computationally efficient implementation of the Cross Stage Partial (CSP) concept that enhances the network's ability to extract rich features with lower overhead. Following the backbone, the neck network is responsible for multi-scale feature fusion. It utilizes an optimized Spatial Pyramid Pooling Fast (SPPF) module to aggregate features from different receptive fields, which is crucial for detecting objects of various sizes. A key innovation in YOLOv11 is the introduction of the C2PSA module. This attention mechanism allows the model to focus more effectively on salient regions within an image, significantly improving its ability to detect small or partially occluded objects. The head then processes these refined feature maps to generate the final bounding box predictions and class probabilities.

The model is pre-trained on real-world DVS recordings and subsequently fine-tuned on our custom CARLA dataset to ensure robust sim-to-real compatibility.

3.2. CARLA DVS Dataset Generation

We generated a comprehensive dataset using the CARLA simulator (V0.9.13) and its Python API. The entire pipeline was built upon the carla-ros-bridge package, allowing for control of the ego-vehicle and the synchronized recording of sensor data. Traffic, including a configurable number of vehicles and pedestrians, was dynamically spawned in the simulation using the provided API scripts. Data was captured by subscribing to corresponding ROS topics, ensuring that each DVS frame has a precisely corresponding RGB frame from the same timestamp. To streamline the annotation process, we first labeled the clear RGB images and then transferred the 2D bounding boxes to the corresponding DVS event images. A critical step was ensuring the sensor resolutions were identical (e.g., 640×640) to guarantee perfect alignment. During data collection, the vehicle is maintained in cruise mode to ensure that each RGB frame has a corresponding DVS reading, since the DVS only produces events when intensity changes occur in the scene (in motion). This approach prevents the creation of empty or invalid bounding boxes when transferring annotations from the RGB dataset to the DVS domain.

To ensure diversity, we recorded scenarios across multiple CARLA built-in maps, as described in Table 2, which cover environments from dense urban layouts to rural country roads. Furthermore, we systematically varied the environmental conditions by programmatically adjusting weather and lighting parameters such as sun altitude angle, cloudiness, precipitation, and fog density. This process resulted in a dataset of 10,716 paired RGB and DVS images.

The complete dataset, comprising 10,716 paired images, is systematically organized for machine learning workflows. We partitioned the data into training, validation, and test sets following a standard 7:2:1 ratio. This division results in 7501 pairs for the training set, 2143 pairs for the validation set, and 1072 pairs for the test set. To foster collaboration and enable the reproducibility of our results, the entire dataset, including images and annotations, will be made publicly available.

Table 2. Summary of CARLA Maps Used for Data Generation.

Town	Summary
Town01	A compact riverside town characterized by multiple bridges and straightforward road layouts.
Town02	A small mixed-use town combining residential neighborhoods with light commercial areas, offering slightly more complexity in traffic flow.
Town03	A larger, urban map with a roundabout and large junctions.
Town07	A rural countryside setting with narrow winding roads, farmland, barns, and minimal traffic infrastructure, emphasizing naturalistic driving conditions.
Town10	A bustling downtown environment with tall skyscrapers, residential districts, and a scenic oceanfront promenade, suited for advanced urban driving challenges.

3.3. Training Details

We trained several model (YOLOv11n, YOLOv11s, and YOLOv11m) to evaluate the trade-off between performance and computational cost. A distinct pre-training strategy was adopted for each modality. For the event-based models, we employed a transfer learning approach, initializing the network weights from a model pre-trained on a large-scale, real-world DVS dataset as established in prior work. This leverages features learned from real event data. In contrast, the RGB models were initialized using the standard pre-trained weights provided with the official YOLOv11 release, which are trained on the COCO dataset. All models were then fine-tuned on our custom CARLA dataset. The training was conducted for 300 epochs with a batch size of 16. We utilized the AdamW optimizer with an initial learning rate of 0.001, which was gradually reduced using a cosine annealing scheduler. The input image resolution was maintained at 640×640 pixels.

3.4. System Setup and Configuration

As detailed in Section 2.5, CARLA's DVS still outputs data in a frame-based manner rather than truly asynchronous events. To approximate the physical characteristics of event cameras while keeping consistency in the experimental setup, both the DVS and RGB cameras are configured to the same resolution. However, the DVS sampling frequency is set to 120 Hz, which is ten times higher than the RGB camera frequency. This 10:1 ratio is a pragmatic compromise based on a clear trade-off. In reality, event cameras can achieve a temporal advantage equivalent to a 10:1 to 40:1 ratio over standard 30–120 FPS automotive cameras. Replicating the upper end of this range is computationally intractable within the synchronous, frame-based CARLA simulator. We, therefore, selected this 10:1 ratio as a conservative, yet computationally feasible, baseline. It provides a reasonable approximation that fairly demonstrates the core temporal advantage of DVS without creating an exaggerated comparison. Importantly, simply using a higher frame-rate RGB camera is not directly comparable, as it would produce redundant full-frame data with higher bandwidth and computational cost, while still lacking the dynamic range and microsecond latency benefits intrinsic to event cameras.

Figure 1 presents a comparative timeline illustrating the data processing and detection flow for two distinct sensing modalities. The top pipeline (RGB Camera) depicts a conventional system operating at a low frequency. It captures discrete, complete frames at fixed intervals (e.g., 20–30 Hz). Consequently, object detection can only occur at these sparse moments, creating significant temporal gaps where critical information may be missed. The bottom pipeline (Event Camera) represents our proposed approach. It receives a continuous, dense stream of asynchronous event data. For processing, this stream is partitioned into "Sliced Event Frames" at a much higher frequency (120 Hz in our simulation). This is also exactly how event frame is simulated in CARLA. This results in a far denser series of

detection opportunities, dramatically reducing the latency between a real-world event and its digital detection in theory.

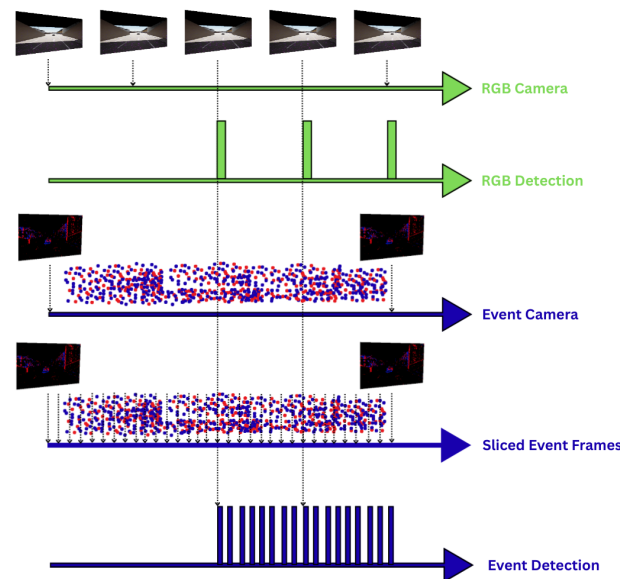


Figure 1. Illustration of the temporal advantage of the DVS system, where the blue and red (positive and negative) dots represents continuous event stream .

Note that in practice, the key benefit of this structure is not in making every single TTC calculation faster, but in enabling the first critical TTC calculation to happen much sooner. The AEB trigger requires a synchronized pair of data: a bounding box (what to look at) and a depth map (how far away it is). With the event-based system, when objects appear, the event camera immediately registers the motion. A new “sliced event frame” is generated within milliseconds, and a detection is made almost instantly. The system can then take this new, low-latency bounding box and pair it with the most recently available depth map from the range sensor that may be operating at a much lower frequency.

3.5. Fusion Technique

As discussed in Section 2.4, multi-sensor fusion strategies can generally be classified into feature-level and proposal-level approaches. Our AEB pipeline adopts a proposal-level fusion strategy, which performs object detection followed by depth sampling for two key reasons aligned with the system’s design objectives. First, in terms of real-time efficiency, feature-level fusion, although capable of producing richer joint representations, often incurs substantial computational overhead. In contrast, our proposal-level design remains computationally lightweight by restricting depth processing to the localized regions of interest identified by YOLOv11, thereby preserving the stringent timing requirements of the AEB control loop. Secondly, the approach promotes modularity and robustness by decoupling detection (YOLOv11) from ranging (LiDAR) tasks. This modular structure aligns naturally with our ROS-based architecture, enabling sensors to operate at different frequencies (e.g., 120 Hz for DVS versus lower-rate depth acquisition) and reducing sensitivity to calibration inaccuracies that often hinder complex feature-level fusion methods. While this strategy inherently omits potential threats outside the detected regions, it remains sufficient for the objectives of this study, which focus on demonstrating the feasibility of real-time event-based perception-to-control integration with well-distinguished object classes.

3.6. Detection-to-Control Pipeline

This section details the implementation of the integrated perception and control pipeline for the AEB system of the ego vehicle. The system is designed as a node within the Robot Operating System (ROS) framework through CARLA-ROS-Bridge, enabling modularity and robust real-time data handling. The core function of this pipeline is to process perception data from a vision-based detection system, assess collision risk, and execute appropriate longitudinal control commands—either maintaining a cruise speed or initiating emergency braking.

3.6.1. System Architecture and Data Acquisition

The system subscribes to three primary data streams within the CARLA simulation environment:

1. **Object Bounding Boxes:** 2D bounding box coordinates are received from a YOLOv11 object detection node. These boxes identify the spatial location of potential obstacles in the camera frame.
2. **Depth Data:** The depth information is derived from simulated LiDAR measurements at 20 Hz and reformatted into a per-pixel depth map representation, where each pixel encodes the distance from the sensor to the corresponding point in the scene.
3. **Ego-Vehicle Velocity:** The current longitudinal velocity of the ego-vehicle, v_{ego} , is obtained from the vehicle's speedometer data.

A key architectural design choice involves the data synchronization strategy. To ensure that distance measurements are calculated using spatially and temporally consistent data, the bounding box and depth image topics are synchronized. This synchronizer matches messages with the closest timestamps, accounting for minor latencies in the perception pipelines. In contrast, the ego-vehicle's velocity is handled by a separate, non-synchronized callback. This decouples the control state from the perception pipeline, ensuring that the most recent velocity measurement is always used for risk assessment, which is critical for accurate TTC calculation.

3.6.2. Collision Risk Assessment via Time-to-Collision

The primary metric for collision risk assessment is the TTC. The calculation process is triggered upon receiving a synchronized pair of a bounding box and a depth image. First, the system extracts the Region of Interest (ROI) from the depth image that corresponds to the provided bounding box coordinates. To enhance robustness, the effective distance to the obstacle, D , is determined by calculating the 5th percentile of all valid (non-zero) depth values within this ROI. This statistical approach is a deliberate safety choice. Using the 1st percentile (absolute minimum distance) would make the system highly susceptible to sensor noise, potentially causing false-positive braking. Conversely, using the mean or median is highly dangerous, as a bounding box often includes background pixels (e.g., the road visible around a pedestrian). This would average the true object distance with the farther background, overestimating D and dangerously delaying the AEB trigger. The 5th percentile provides a robust, conservative estimate of the object's closest surface while effectively filtering out the bottom 5% of potential noise outliers, thus stabilizing the TTC calculation.

The TTC is then calculated using the standard formula:

$$TTC = \frac{D}{v_{ego}} \quad (2)$$

where D is the effective distance to the obstacle (in meters), and v_{ego} is the current longitudinal speed of the ego-vehicle (in meters per second). This calculation is only performed if

v_{ego} is greater than a nominal threshold, to prevent division by zero or near-zero values when the vehicle is stationary.

3.6.3. Dual-Mode Longitudinal Control Logic

The vehicle's longitudinal control operates in one of two modes: In the absence of an immediate collision threat, the system maintains a user-defined target speed. A Proportional-Integral-Derivative (PID) longitudinal controller is employed for this purpose. The controller calculates the necessary throttle output by comparing the target speed with the current speed, v_{ego} . The AEB mode is activated if the calculated TTC falls below a predefined safety threshold of 2.0 s. Upon activation, the system immediately overrides the PID controller's output. It commands zero throttle and applies maximum braking. To ensure the vehicle comes to a complete and safe stop without intermittent control oscillations, the AEB state is latched once triggered. It remains active, applying full braking force, until the vehicle's velocity is reduced to near zero, at which point the system concludes its operation. It should be noted that this control logic is intentionally simplified to ensure a decisive and safe stop within our controlled test scenarios, serving primarily to demonstrate the detection-control pipeline rather than a fully developed control algorithm. This behavior is not intended to represent a real-world implementation.

3.7. Scenario Design in CARLA Simulation

To evaluate the proposed system under diverse and safety-critical conditions, we implement three AEB test scenarios that includes illumination-variation and sudden pedestrian-encounter. These scenarios are motivated by the fact that real-world AEB triggering often fails not only due to limitations in perception accuracy but also due to the challenges imposed by changing environments, sudden appearances of vulnerable road users, and highly dynamic traffic interactions. Some key parameters are included in Table A1 for reproducibility.

1. Tunnel Exit/Sudden Illumination Change (Stationary): In this scenario, the ego vehicle exits a dark tunnel into bright daylight. Such transitions induce strong illumination variations that challenge conventional frame-based sensors, often leading to temporary blindness or delayed response. Event-based sensors, with their high dynamic range, are particularly well-suited to this condition, making it an essential test case for assessing robustness against abrupt lighting changes. Within CARLA simulator, the exposure mode of RGB camera was changed from 'auto' to 'histogram' and exposure speed was adjusted to produce the glaring effect as shown in Figure 2b.

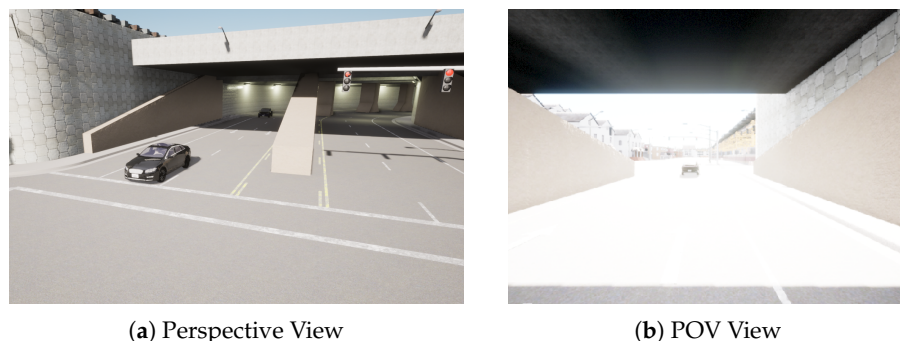


Figure 2. Tunnel Exit Scenario where the glaring effect is 'blinding' the RGB camera.

2. Poor Illumination: Night Driving (Stationary): The second stationary scenario places the ego vehicle on a rural road at night, as shown in Figure 3. Poor illumination is a well-documented failure mode of camera-based perception systems, where object visibility and depth perception degrade significantly. This setup enables us to

test whether the proposed integration of event-based detection and depth cues can maintain reliable performance under extremely low-light conditions.



Figure 3. Low-Illumination Scenario where vehicle in front is not visible in RGB camera.

3. **Pedestrian Crossing (Dynamic):** This is a very typical and challenging dynamic scenario, a pedestrian suddenly emerges from behind an obstacle, leaving minimal reaction time for the ego vehicle, as shown in Figure 4. This scenario is designed to test semantic filtering of safety-critical objects, particularly vulnerable road users (VRUs), which represent high-risk events for AEB systems. Robust detection under such occlusion and sudden appearance conditions is crucial for ensuring safety.



Figure 4. Obstructed Pedestrian Crossing Scenario.

Together, these scenarios cover a representative spectrum of environmental and interaction challenges: abrupt illumination transitions, extreme low-light driving, sudden pedestrian appearances situations. They allow us to systematically assess both the semantic reliability and the temporal responsiveness of the proposed system, while reflecting real-world conditions where AEB performance directly impacts safety.

4. Results

4.1. Detection Model Results

The performance of the YOLOv11 model series was evaluated using data from both an event camera and a standard RGB camera. The quantitative results for both modalities are presented to compare their effectiveness for the object detection task.

Table 3 demonstrates that RGB camera data consistently outperforms event camera data across all models. The YOLOv11m model achieves the highest overall mAP@.50-.95 of 0.886 on RGB data, representing a 25.1% improvement over the 0.708 achieved on event camera data. YOLOv11s exhibits the best precision at 0.959. As model complexity increases, performance shows steady improvement across both modalities.

Table 3. Overall Detection Performance Comparison.

Model	Data Type	Precision	Recall	mAP@.50	mAP@.50-.95
YOLOv11n	Event Camera	0.922	0.856	0.930	0.695
	RGB Camera	0.955	0.940	0.985	0.869
YOLOv11s	Event Camera	0.923	0.876	0.943	0.702
	RGB Camera	0.959	0.935	0.985	0.884
YOLOv11m	Event Camera	0.920	0.862	0.937	0.708
	RGB Camera	0.949	0.949	0.983	0.886

Bold values indicate the best performance in that column.

Table 4 shows that vehicle detection outperforms pedestrian detection on both data types. On RGB data, YOLOv11m achieves the optimal mAP@.50-.95 of 0.922, representing a 17.6% improvement over the 0.784 achieved on event camera data. On event camera data, YOLOv11s achieves the highest vehicle detection mAP@.50-.95 of 0.788. As larger and more distinctive objects, vehicles are well-detected across both data modalities.

Table 4. Vehicle Detection Performance Comparison.

Model	Data Type	Precision	Recall	mAP@.50	mAP@.50-.95
YOLOv11n	Event Camera	0.933	0.910	0.961	0.774
	RGB Camera	0.952	0.968	0.989	0.909
YOLOv11s	Event Camera	0.942	0.917	0.969	0.788
	RGB Camera	0.961	0.960	0.989	0.921
YOLOv11m	Event Camera	0.946	0.900	0.965	0.784
	RGB Camera	0.955	0.967	0.988	0.922

Bold values indicate the best performance in that column.

Table 5 reveals that pedestrian detection presents a more challenging task. The performance improvement on RGB data is most significant, with YOLOv11m's mAP@.50-.95 increasing from 0.632 on event camera data to 0.850 on RGB data—a 34.5% improvement, the largest across all categories. On event camera data, YOLOv11m's pedestrian detection mAP@.50-.95 is only 0.632, indicating that sparse event data poses greater challenges for detecting small targets and pedestrians with varying poses.

Table 5. Pedestrian Detection Performance Comparison.

Model	Data Type	Precision	Recall	mAP@.50	mAP@.50-.95
YOLOv11n	Event Camera	0.911	0.801	0.899	0.616
	RGB Camera	0.959	0.913	0.981	0.830
YOLOv11s	Event Camera	0.904	0.836	0.916	0.616
	RGB Camera	0.958	0.911	0.981	0.847
YOLOv11m	Event Camera	0.895	0.824	0.908	0.632
	RGB Camera	0.944	0.931	0.978	0.850

Bold values indicate the best performance in that column.

- **Advantages of RGB Images:** RGB images provide dense, feature-rich information including color, texture, and brightness gradients. Convolutional neural networks are naturally suited to this grid-like data structure, showing significant advantages across all detection tasks, particularly for small targets such as pedestrians.
- **Characteristics of Event Camera Data:** Event camera data is sparse, capturing only brightness changes. While offering advantages in temporal resolution and power consumption, it presents greater challenges for object detection tasks using tradi-

tional convolutional architectures. The reduced static feature information requires specialized processing methods to fully leverage its benefits.

It is essential to frame these mAP results in the context of our paper's primary objective. While Table 3 demonstrates that the RGB model achieves a higher general-purpose mAP across the entire dataset, this single metric does not capture performance in the most safety-critical, high-risk scenarios.

The core value proposition of DVS is not to outperform RGB in all conditions, but to provide critical robustness in specific, challenging environments (like the high-glare tunnel exit) where conventional sensors are known to fail. This crucial trade-off—sacrificing some general-case accuracy for a decisive gain in edge-case robustness—is the focus of our system-level evaluation.

4.2. AEB Scenario Results

To quantitatively assess the performance of our proposed event-based AEB system, we established a comprehensive set of performance metrics and two distinct baseline systems for comparative analysis. This methodology is designed to rigorously evaluate the system's effectiveness from perception through to final control action and to isolate the specific advantages conferred by the use of an event-based vision sensor.

4.2.1. Baseline for Comparison

Two baseline systems were implemented to provide a robust comparative context for our results:

- **RGB-Integrated System:** This serves as our primary baseline and features an identical software architecture to our proposed system. However, it replaces the event camera with a conventional RGB camera, processing standard image frames through the YOLOv11 [30] object detector (m size model used in both RGB and event model). This configuration allows for a direct, modality-focused comparison, isolating the performance differences attributable solely to the choice of vision sensor (Event vs. RGB), particularly under challenging environmental conditions.
- **Depth-Only System:** This baseline completely omits the object detection module. Instead, it triggers an AEB response based on a simple distance threshold applied to any point within the forward-looking depth map generated from a LiDAR running at 20Hz. The purpose of this baseline is to validate the critical role of the perception front-end in distinguishing between genuine threats (e.g., pedestrians, other vehicles) and irrelevant objects (e.g., overhead signs, roadside clutter), thereby highlighting the necessity of object detection for minimizing false positives.

4.2.2. Performance Metrics

To provide a comprehensive evaluation, a structured set of metrics was established to assess distinct aspects of the AEB system's performance within the scenarios, which include perception accuracy and latency, overall system safety, and real-world reliability.

The initial stage of evaluation focuses on Perception Performance, which is quantified by two key metrics. First, the Detection Rate (Recall) serves as the primary measure of perception accuracy, defined as the ratio of correctly identified threats to the total number of actual threats present across multiple runs. Complementing this, the Time to First Reliable Detection (TFRD) measures the system's perception sensitivity. Reliable detection is defined as the elapsed time from the start of the scenario to the moment an obstacle is classified as a potential threat, quantifying the system's early detection capability. In practice, 'reliable' refers to consistent sensor observations over consecutive readings to reduce false positives.

Moving beyond perception, system-level safety and effectiveness are assessed to determine the success of the entire pipeline. The principal metric here is the Collision Avoidance Rate, which is the percentage of test runs where the AEB system successfully brings the vehicle to a stop without obstacle contact. The system reaction time is measured by the Time to Activation (TTA), which captures the total duration from the start of test runs to the application of the brakes, thus incorporating both perception latency (TFRD) and subsequent decision-making and actuation time.

At last, System Reliability provides another reference to gauge real-world usability. This is quantified by the False Positive Rate, measured as the number of erroneous AEB activations during test runs where no collision threat exists. A low false positive rate is critical for ensuring operational safety and maintaining driver trust.

4.2.3. Test Results from Scenario

- **Tunnel Exit/Sudden Illumination Change** In the tunnel exit scenario, a stationary vehicle is spawned 100 m ahead of the ego vehicle that is cruising at 40 km/h. Total attempts of tests is 30 and the cruise speed varies within 5% to create variation. The results are shown in Table 6.

Table 6. Comparison of Performance for Different Sensor Modalities in Tunnel Scenario (N = 30 Trials).

Sensor Modality	Perception Performance			System Effectiveness	
	TFRD (s)	Recall Rate	False Positive Rate	TTA (s)	Collision Avoidance Rate
DVS + Depth	Mean: 0.775	100%	0%	Mean: 2.026	100%
	Std: 0.054			Std: 0.080	
RGB + Depth	Mean: 1.632	13.3%	0%	Mean: 3.150	0%
	Std: 0.042			Std: 0.067	
Depth Only	Mean: 1.860	100%	0%	Mean: 2.571	86.6%
	Std: 0.140			Std: 0.296	

The AEB system equipped with DVS demonstrated exceptional robustness and efficiency. It achieved a perfect 100% Recall Rate and 100% collision avoidance rate, successfully detecting and avoiding the stationary vehicle in all 30 trials. The stand-out metric is the TFRD, which had a remarkably low mean of 0.775 s with minimal deviation ($\sigma = 0.054$). This rapid detection is a direct result of the DVS's physical property, as it records asynchronous events based on changes in log-scale luminance rather than absolute light intensity. This makes it inherently resilient to the sudden, drastic illumination shift at the tunnel exit, enabling near-instantaneous perception where traditional cameras fail.

In contrast, the performance of the conventional RGB-based AEB system was found to be critically deficient under the tested low-light conditions. The system only managed to detect the obstacle in 13.3% of the trials. Even in the few instances where detection occurred, the average detection reaction times was 2.632 s, which were substantially longer than those of the alternative sensor modalities. This delay is caused by the RGB camera's auto-exposure algorithm struggling to adapt from the tunnel's darkness to the bright exterior, effectively "blinding" the system at the critical moment. Consequently, by the time the system could process the scene and trigger an avoidance action (mean TTA of 3.150 s), it was far too late to stop the vehicle, resulting in 0% collision avoidance rate.

The Depth-Only configuration also performed well, proving to be a highly reliable perception tool. It successfully detected the stationary vehicle in all 30 trials and

achieved a high Collision Avoidance Rate of 86.6%. As an active sensor, the depth camera provides its own illumination to measure distance, rendering its function independent of ambient light. This explains its perfect recall rate. However, its detection time was significantly slower than the DVS system, with a mean TFRD of 1.860 s—more than double that of the DVS. This latency is likely attributable to the difference in operating frequencies mentioned in the simulation setup (20 Hz for depth vs. a much higher frequency for DVS). While highly reliable, this slower detection time was not sufficient to guarantee collision avoidance in every single trial, explaining the gap between its 100% detection rate and 86.6% avoidance rate.

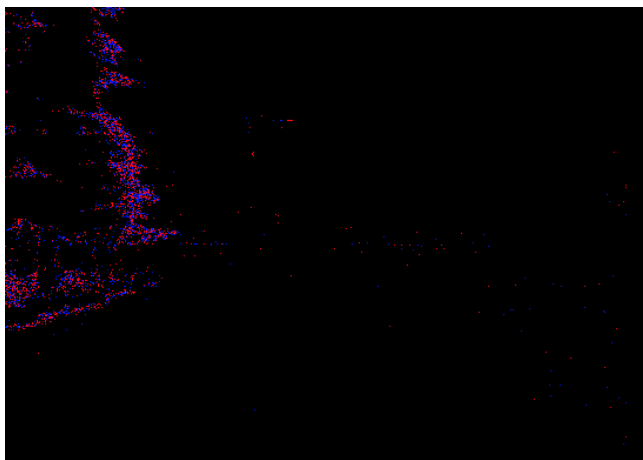
As AEB systems operate on millisecond timescales, performance differences may not be immediately apparent from visual inspection alone. To quantitatively assess the significance of these differences, two-sample t-tests were conducted on TFRD and TTA between the DVS-based and Depth-based systems. The RGB system was excluded due to insufficient sample size, as it was largely nonfunctional under the tested conditions. The resulting near-zero p -values ($p < 0.05$) confirm a statistically significant difference between the two sensor modalities: the DVS system achieves approximately 60% faster detection and 21% quicker braking response compared to Depth sensing. These results highlight the advantage of DVSs for rapid perception and timely reaction, particularly in tunnel-exit scenarios characterized by abrupt illumination changes.

- **Poor Illumination Scenario** In the sustained low-illumination scenario, the performance hierarchy of the sensor modalities shifted dramatically compared to the tunnel-exit test, as shown in Table 7. The Depth-Only system delivered a flawless performance, emerging as the superior solution. The DVS + Depth system's effectiveness was notably degraded, while the RGB + Depth system remained poor performance. The Depth-Only configuration was the undisputed top performer in this scenario, achieving a 100% Recall Rate and a perfect 100% Collision Avoidance Rate. It was also the fastest to react, with a mean Time to First Reliable Detect of just 1.268 s, significantly outperforming the other configurations. The results from independent two-sample t-tests showed that Depth-based system significantly outperformed DVS in both detection and AEB response times in this scenario. This exceptional result is due to its nature as an active sensor. By emitting its own infrared light to measure distance, its perception is completely independent of ambient environmental light. This makes it inherently robust and reliable for operation in consistently dark conditions, such as nighttime driving. The DVS-based system, which excelled in the dynamic lighting change scenario, showed some level of performance degradation in static low light. Its Collision Avoidance Rate dropped to 73.3%, and its Recall Rate fell to 86.6%. Additionally, this configuration introduced a 6.7% False Positive Rate, indicating that the system occasionally detected obstacles that were not there.

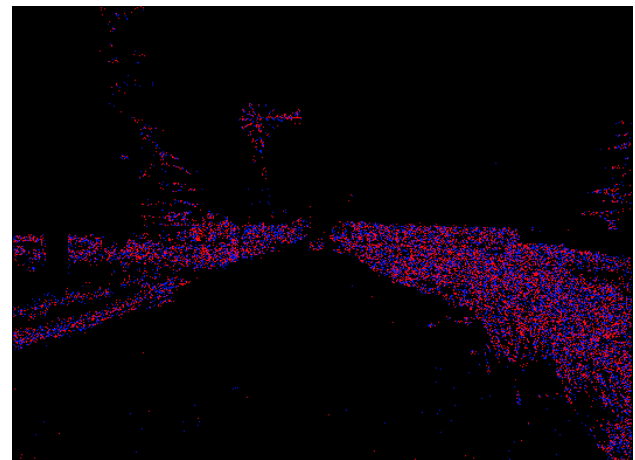
This decline is fundamentally rooted in the sensor's vulnerability to a poor signal-to-noise ratio in the dark, a problem inherent in the DVS model within the CARLA simulator itself, as shown in Figure 5. Visual data shows that while the daytime feed consists of clean, sparse events driven by actual motion, the nighttime feed is overwhelmed by a dense cloud of spurious background activity. This simulated electronic noise floor effectively masks the weak event-based signal generated by the actual target vehicle. The perception algorithm is consequently forced to distinguish a subtle pattern from a chaotic background, which directly explains the observed results: a slower reaction time (2.301 s TFRD) as the system struggles to gain confidence, a lower recall rate from missing the masked signal, and the introduction of a 6.7% False Positive Rate caused by mistaking random noise clusters for real objects.

Table 7. Comparison of Performance for Different Sensor Modalities in Low-Illumination Scenario (N = 30 Trials).

Sensor Modality	Perception Performance			System Effectiveness	
	TFRD (s)	Recall Rate	False Positive Rate	TTA (s)	Collision Avoidance Rate
DVS + Depth	Mean: 2.301	86.6%	6.7%	Mean: 2.342	73.3%
	Std: 0.282			Std: 0.289	
RGB + Depth	Mean: 1.607	33.3%	0%	Mean: 1.646	0%
	Std: 0.074			Std: 0.091	
Depth Only	Mean: 1.268	100%	0%	Mean: 2.038	100%
	Std: 0.030			Std: 0.075	



(a) DVS in day



(b) DVS at night

Figure 5. Comparison of the same scenario under different noise levels: (b) exhibits noticeably higher noise than (a).

The RGB-based system continued to demonstrate a critical failure, resulting in a 0% Collision Avoidance Rate. This was a direct consequence of its severely compromised perception, which achieved a meager 33.3% Recall Rate. As a passive sensor, the RGB camera's effectiveness is entirely dependent on sufficient ambient light. In the low-illumination conditions, images were too dark and noisy for the vision algorithm to detect the obstacle at a safe distance. The few successful detections can be attributed to moments at very close proximity when the ego vehicle's own headlights finally illuminated the obstacle. However, this critically late detection left insufficient time for the AEB system to engage, leading to an unavoidable collision in every trial. Therefore, even with its noted degradation in low light, the DVS camera's perception remained far superior to that of the RGB camera, which was rendered almost entirely non-functional.

- **Pedestrian Crossing** A much more critical scenario was introduced for the pedestrian crossing test, where the threat appears only 15 m in the vehicle's path while it cruises at 25 km/h. It again reshapes the performance hierarchy, as shown in Table 8. In this test, the RGB + Depth system emerged as the most effective solution, yet both the DVS + Depth and Depth-Only systems struggled significantly, revealing critical weaknesses in their perception capabilities for this specific task.

Table 8. Comparison of Performance for Different Sensor Modalities in Pedestrian Crossing Scenario (N = 30 Trials).

Sensor Modality	Perception Performance			System Effectiveness	
	TFRD (s)	Recall Rate	False Positive Rate	TTA (s)	Collision Avoidance Rate
DVS + Depth	Mean: 0.761	13.3%	6.7%	Mean: 1.125	13.3%
	Std: 0.072			Std: 0.080	
RGB + Depth	Mean: 1.607	100%	0%	Mean: 1.646	73.3%
	Std: 0.074			Std: 0.091	
Depth Only	Mean: 1.228	33.3%	36.6%	Mean: 2.080	50%
	Std: 0.140			Std: 0.296	

The RGB-based system excelled in perception, achieving a perfect 100% Recall Rate. This success stems from its ability to leverage rich semantic clues; trained on extensive datasets, the vision algorithm reliably identifies the features and form of a human, an advantage that is crucial for smaller object detection. However, its perfect detection did not translate to perfect safety, resulting in a 73.3% Collision Avoidance Rate. The primary limiting factor was its reaction speed. The detection pipeline's low operating frequency resulted in a mean Time to First Reliable Detect of 1.607 s. This latency is inadequate for time-critical edge scenarios like this one, where every millisecond is crucial for a successful outcome. This created a critical gap where the system saw the threat but could not trigger the AEB in time to prevent a collision, demonstrating that detection without sufficient rate might not be enough.

The Depth-Only system's performance degraded alarmingly, with a low 33.3% Recall Rate and an extremely high 36.6% False Positive Rate. This configuration struggled to reliably distinguish the pedestrian's geometric profile from background clutter like poles or signs. This constant mismatching of threats reveals the fundamental limitation of using raw depth data without semantic context. The sensor can detect a shape but has no understanding of what that shape is, leading to both missed detections and frequent false alarms. This severe unreliability powerfully justifies the need for detection-level integration in any AEB system. Fusing this depth data with a semantic sensor like an RGB camera would provide the necessary context to confirm threats and reject false positives, drastically improving system safety.

The DVS-based system proved largely ineffective for this task, with a 13.3% Recall Rate and a corresponding 13.3% Collision Avoidance Rate. This failure highlights a key challenge for event-based detection that is more complex than simple sparseness. The RGB model's 100% recall demonstrates its clear semantic advantage: it is trained to recognize the shape and form of a human, which is a consistent feature. The DVS model, by contrast, must recognize a motion signature. A pedestrian's non-rigid, articulated motion (e.g., swinging arms and legs) creates a complex, variable, and less coherent event cluster compared to the sharp, predictable, moving edges of a rigid vehicle. Furthermore, in this dynamic scenario, the ego-vehicle's own forward motion generates a dense stream of background events (optic flow) from static objects like the curb and scenery. The DVS model must therefore distinguish the pedestrian's event signal from this background motion noise. This "signal-to-noise" problem is absent for the RGB camera, which easily separates a semantic human shape from a static background, regardless of ego-motion.

However, the underlying strength of the DVS is still evident. In the few trials where it successfully detected the pedestrian, its TFRD was an exceptional 0.761 s, confirming

its temporal advantage remains intact. The 100% detection-to-avoidance conversion in these cases further validates this advantage. Therefore, the challenge is not the speed of the data, but the fundamental difficulty of robustly interpreting a non-rigid object's motion signature amidst a noisy background of optic flow—a task that our current CNN-based model struggled with. A series of approaches in the processing pipeline could influence the final result for this case. Our specific implementation, which relies on a CNN-based detector, a basic frame-stacking strategy, and no dedicated de-noising method, struggled with this task. Exploring alternative approaches, as discussed in our future work, will be critical to overcoming this limitation.

4.3. Real-Time Performance and Deployment Feasibility

A key requirement for any practical AEB system is the ability to execute the entire perception pipeline in real-time, matching the high-frequency data from the sensors. Our pipeline's DVS operates at 120 Hz, which demands a total processing time of approximately 8.3 milliseconds per frame. In our experimental setup with NVIDIA GeForce RTX 4060 Ti, we need to make sure the YOLO inference time is not the bottleneck of the overall system. Expanding on that, it is also essential to confirm that our chosen architecture (YOLOv11) is feasible for real-world deployment on typical automotive-grade (edge) hardware.

To demonstrate this, we present a benchmark analysis of our YOLOv11 model, deployed on various NVIDIA Jetson platforms, which are representative of hardware used in autonomous systems. The benchmark, detailed in Table 9, evaluates the model (640×640 input) using both standard PyTorch and an optimized TensorRT INT8 configuration.

Table 9. Performance comparison across different devices and setups.

Device (Full Name)	Setup	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total (ms)	Min (ms)	Max (ms)
NVIDIA GeForce RTX 4060 Ti	PyTorch	1.02 ± 0.21	4.33 ± 0.40	0.94 ± 0.47	6.25 ± 0.54	2.5	8.2
NVIDIA Jetson Orin NX 16 GB	PyTorch	1.34 ± 0.27	11.8 ± 0.9	1.07 ± 0.53	14.21 ± 1.15	11.6	17.9
NVIDIA Jetson Orin NX 16 GB	TensorRT INT8	1.28 ± 0.24	4.9 ± 0.5	1.12 ± 0.49	7.30 ± 0.83	5.9	9.2
NVIDIA Jetson Orin Nano 8 GB	PyTorch	1.40 ± 0.28	18.5 ± 1.5	1.15 ± 0.57	21.05 ± 1.78	17.3	25.6
NVIDIA Jetson Orin Nano 8 GB	TensorRT INT8	1.32 ± 0.26	7.6 ± 0.7	1.18 ± 0.56	10.10 ± 0.95	8.3	12.9
NVIDIA Jetson AGX Orin 32 GB	PyTorch	1.22 ± 0.24	6.9 ± 0.6	1.13 ± 0.51	9.25 ± 0.92	7.8	11.6
NVIDIA Jetson AGX Orin 32 GB	TensorRT INT8	1.27 ± 0.23	2.8 ± 0.3	1.06 ± 0.48	5.13 ± 0.59	4.2	6.6

The benchmark results serve two key purposes. First, the data from our NVIDIA GeForce RTX 4060 Ti shows an average total processing time of 6.25 ms. This is well below our 8.3 ms budget, confirming that YOLO inference was not a performance bottleneck during our simulation-based experiments.

Furthermore, the data from the edge devices validates our approach's real-world deployment feasibility. While a standard PyTorch implementation is too slow (e.g., 14.21 ms on Orin NX), a properly optimized model using TensorRT INT8 quantization achieves significant $1.8\times$ to $2.1\times$ speedups. Crucially, the optimized latency on the Jetson AGX Orin 32 GB (5.13 ms) and the Jetson Orin NX 16 GB (7.30 ms) are both below the 8.3 ms processing budget required by our 120 Hz pipeline. This analysis confirms that the computational demands of our proposed event-based YOLOv11 pipeline are not just theoretical but are practically achievable on current, commercially available edge hardware.

5. Conclusions

This paper presented the design, integration, and closed-loop validation of a complete AEB pipeline. Our primary contribution is the system-level demonstration of this

pipeline, which uniquely integrates a state-of-the-art YOLOv11-based detector, a conservative depth-fusion logic, and a real-time ROS control loop to leverage the benefits of DVS. To validate this system, we developed and released a novel two-channel CARLA DVS dataset and a benchmark of challenging AEB scenarios designed to test known failure modes of conventional sensors.

Our comprehensive evaluation confirms the central hypothesis of this work: event-based sensors are a powerful solution for specific, safety-critical scenarios where conventional RGB cameras fail. The DVS pipeline's high temporal resolution and inherent HDR capabilities provided decisive, rapid perception in the high-glare tunnel exit scenario. However, our results also show there is no single superior sensor modality for all conditions. The DVS system struggled to reliably interpret the sparse event signal from pedestrians, a task where the semantic-rich RGB sensor excelled. Ultimately, this research validates the significant potential of event cameras not as a universal replacement, but as a powerful complementary component in an automotive sensor suite. Their core advantage lies in providing extreme robustness and low-latency perception in specific, high-risk scenarios where conventional systems are known to fail.

6. Limitations and Future Work

While our work demonstrates a successful integration, we acknowledge several limitations that provide clear directions for future research as direct extensions of our established work:

Control Logic: The AEB control logic was a simple “latching” mechanism that applied maximum braking until the vehicle stopped. This simplification was used for controlled test scenarios and is not suitable for real-world applications where a threat might be clear. Future work could implement a more advanced controller that continuously re-evaluates the TTC and can de-latch the brakes when appropriate.

Detection Algorithm and Fusion Techniques: Our methodology employed a frame-based detector (YOLOv11) processing “Sliced Event Frames.” While practical for our ROS-based system and computationally efficient, this approach does not fully exploit the microsecond-level temporal resolution of raw event data. A valuable future comparison would be to benchmark this pipeline against asynchronous, recurrent architectures (such as RED or RVT) to quantify the trade-offs between system practicality and raw latency. Also, pairing the RGB camera and DVS camera has becoming a promising approach in autonomous driving systems. Therefore another direction is implementing and evaluating a feature-level fusion model that combines DVS, RGB and depth data at a lower level, which could improve robustness at the cost of computational complexity and tighter sensor synchronization.

Sensor Modeling: The CARLA sensor's contrast threshold was fixed at $C = 0.3$. This hyperparameter directly affects the event stream's density and signal-to-noise ratio, as illustrated in the low-illumination scenario. A comprehensive sensitivity analysis of this threshold, along with improved dataset labeling techniques, is a necessary next step for enhancing model robustness.

Author Contributions: Conceptualization, J.F., P.K., J.K. and A.S.; methodology, J.F., P.K. and P.Z.; software, J.F. and P.Z.; validation, J.F., P.Z. and H.Z.; formal analysis, J.F. and P.Z.; investigation, J.F., J.K. and A.S.; resources, J.K., A.S. and P.K.; data curation, J.F. and P.Z.; writing—original draft preparation, J.F.; writing—review and editing, J.F., P.Z., J.K., A.S., H.Z. and J.W.; visualization, J.F.; supervision, J.K., A.S. and P.K.; project administration, J.K. and A.S.; funding acquisition, J.K. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The YOLO detection dataset is currently available on request from the corresponding author and will be publicly available shortly.

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

Abbreviations

The following abbreviations are used in this manuscript:

AEB	Automatic Emergency Braking
DVS	Dynamic vision sensor
YOLO	You Only Look Once
RNN	Recurrent neural network
HDR	High Dynamic Range
LiDAR	Light detection and ranging
CNN	Convolutional neural network
SNN	Spiking neural network
DNN	Dense neural network
GNN	Graph neural network
ROS	Robot Operating System
VRU	Vulnerable road users
TFRD	Time to first reliable detection
TTA	Time to activation
CSP	Cross Stage Partial
C2PSA	Convolutional block with Parallel Spatial Attention

Appendix A

The Appendix provides some key parameters to replicate the testing scenarios presented in the main paper. It is important to note that these configurations represent discrete points within a vast operational domain. The performance of AEB applications is known to vary significantly under different initial speed and distance combinations.

Table A1. Scenario configurations for AEB evaluation in CARLA simulation.

Scenario	Set Ego Speed	Obstacle Type	Initial Distance	Key Configuration(s)
1. Tunnel Exit	40 km/h ($\pm 5\%$)	Stationary Vehicle	100 m	CARLA Setting: CARLA MAP = Town03 sun_altitude_angle = 150 sun_azimuth_angle = 180 Vehicle Camera: "exposure_mode": "histogram" "exposure_compensation": 0.0 "exposure_min_bright": 6.0 "exposure_max_bright": 12.0 "exposure_speed_up": 0.01 "exposure_speed_down": 1.0
2. Low Illumination	40 km/h ($\pm 5\%$)	Stationary Vehicle	100 m	CARLA MAP = Town07 sun_altitude_angle = 0 (Night/Early Dawn)
3. Pedestrian Crossing	25 km/h ($\pm 5\%$)	Dynamic Pedestrian	20 m (at emergence)	CARLA MAP = Town02 Default (standard daylight)

References

1. Sivaraman, S.; Trivedi, M.M. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1786. [\[CrossRef\]](#)
2. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.; Mutz, F.; et al. Self-driving cars: A survey. *arXiv* **2019**, arXiv:1901.04407. [\[CrossRef\]](#)
3. Liu, F.; Lu, Z.; Lin, X. Vision-based environmental perception for autonomous driving. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2023**, *239*, 39–69. [\[CrossRef\]](#)

4. Chen, G.; Cao, H.; Conradt, J.; Tang, H.; Rohrbein, F.; Knoll, A. Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception. *IEEE Signal Process. Mag.* **2020**, *37*, 34–49. [[CrossRef](#)]
5. Li, J.; Dong, S.; Yu, Z.; Tian, Y.; Huang, T. Event-Based Vision Enhanced: A Joint Detection Framework in Autonomous Driving. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1396–1401.
6. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Tabbara, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 154–180. [[CrossRef](#)] [[PubMed](#)]
7. Shaw, D.C.; Shaw, J.Z. Vehicle Collision Avoidance System. U.S. Patent 5,529,138, 25 June 1996.
8. Widman, G.; Bauson, W.A.; Alland, S.W. Development of collision avoidance systems at Delphi Automotive Systems. In Proceedings of the International Conference on Intelligent Vehicles, Stuttgart, Germany, 28–30 October 1998; Citeseer: Berkeley, CA, USA, 1998; pp. 353–358.
9. Gehrig, D.; Scaramuzza, D. Low-Latency Automotive Vision with Event Cameras. *Nature* **2024**, *629*, 1034–1040. [[CrossRef](#)] [[PubMed](#)]
10. Perot, E.; de Tournemire, P.; Nitti, D.; Masci, J.; Sironi, A. Learning to detect objects with a 1 megapixel event camera. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 6–12 December 2020.
11. Gehrig, M.; Scaramuzza, D. Recurrent Vision Transformers for Object Detection with Event Cameras. *arXiv* **2023**, arXiv:2212.05598. [[CrossRef](#)]
12. Jiang, Z.; Xia, P.; Huang, K.; Stechele, W.; Chen, G.; Bing, Z.; Knoll, A. Mixed Frame-/Event-Driven Fast Pedestrian Detection. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8332–8338.
13. Feng, J.; Zhao, P.; Zheng, H.; Konpang, J.; Sirikham, A.; Kalnaowakul, P. Enhancing Autonomous Driving Perception: A Practical Approach to Event-Based Object Detection in CARLA and ROS. *Vehicles* **2025**, *7*, 53. [[CrossRef](#)]
14. Forrai, B.; Miki, T.; Gehrig, D.; Hutter, M.; Scaramuzza, D. Event-based agile object catching with a quadrupedal robot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 12177–12183.
15. Falanga, D.; Kleber, K.; Scaramuzza, D. Dynamic obstacle avoidance for quadrotors with event cameras. *Sci. Robot.* **2020**, *5*, eaaz9712. [[CrossRef](#)] [[PubMed](#)]
16. Zhang, X.; Tie, J.; Li, J.; Hu, Y.; Liu, S.; Li, X.; Li, Z.; Yu, X.; Zhao, J.; Wan, Z.; et al. Dynamic obstacle avoidance for unmanned aerial vehicle using dynamic vision sensor. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2023; pp. 161–173.
17. Andersen, K.F.; Pham, H.X.; Ugurlu, H.I.; Kayacan, E. Event-based navigation for autonomous drone racing with sparse gated recurrent network. In Proceedings of the 2022 European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 1342–1348.
18. European New Car Assessment Programme. Euro NCAP AEB Car-to-Car Test Protocol v4.3. 2022. Available online: <https://www.euroncap.com> (accessed on 10 September 2025).
19. Walters, C.; Hadfield, S. Evreflex: Dense time-to-impact prediction for event-based obstacle avoidance. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 1304–1309.
20. Sun, K.; Li, J.; Dai, K.; Liao, B.; Xiong, W.; Zhou, Y. EvTTC: An Event Camera Dataset for Time-to-Collision Estimation. *arXiv* **2025**, arXiv:2412.05053. [[CrossRef](#)]
21. Li, J.; Liao, B.; Lu, X.; Liu, P.; Shen, S.; Zhou, Y. Event-Aided Time-to-Collision Estimation for Autonomous Driving. *arXiv* **2024**, arXiv:2407.07324. [[CrossRef](#)]
22. Wang, H.; Liu, J.; Dong, H.; Shao, Z. A Survey of the Multi-Sensor Fusion Object Detection Task in Autonomous Driving. *Sensors* **2025**, *25*, 2794. [[CrossRef](#)] [[PubMed](#)]
23. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; Han, S. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023.
24. Zhao, Y.; Gong, Z.; Zheng, P.; Zhu, H.; Wu, S. SimpleBEV: Improved LiDAR-Camera Fusion Architecture for 3D Object Detection. *arXiv* **2024**, arXiv:2411.05292.
25. Hu, C.; Zheng, H.; Li, K.; Xu, J.; Mao, W.; Luo, M.; Wang, L.; Chen, M.; Peng, Q.; Liu, K.; et al. FusionFormer: A Multi-Sensory Fusion in Bird’s-Eye-View and Temporal Consistent Transformer for 3D Object Detection. *arXiv* **2023**, arXiv:2309.05257.
26. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
27. Li, Y.; Fan, L.; Liu, Y.; Huang, Z.; Chen, Y.; Wang, N.; Zhang, Z. Fully Sparse Fusion for 3D Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 7217–7231. [[CrossRef](#)] [[PubMed](#)]

28. Xie, Y.; Xu, C.; Rakotosaona, M.-J.; Rim, P.; Tombari, F.; Keutzer, K.; Tomizuka, M.; Zhan, W. SparseFusion: Fusing Multi-Modal Sparse Representations for Multi-Sensor 3D Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023.
29. Tan, K.; Pavan Kumar, B.N.; Chakravarthi, B. How Real Is CARLA's Dynamic Vision Sensor? A Study on the Sim-to-Real Gap in Traffic Object Detection. *arXiv* **2025**, arXiv:2506.13722. [[CrossRef](#)]
30. Jocher, G.; Qiu, J. Ultralytics YOLO11. Version 11.0.0, 2024. Ultralytics. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 28 September 2025).

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.