



OPEN

## Design and research of bridge collision avoidance system based on camera calibration technology and motion detection

Xiaolei Wang, Shichao Wang✉, Zhihao Wei, Ruijiang Ren & Jingshan Huang

Bridge collisions, particularly those involving over-height vehicles, pose significant threats to public infrastructure, economic stability, and human safety. This study presents an intelligent, vision-based Bridge Collision Avoidance System (BCAS) that leverages advanced camera calibration techniques, motion detection algorithms, and real-time risk assessment frameworks to proactively detect and mitigate potential collisions. The system architecture integrates high-resolution video feeds with precise intrinsic and extrinsic camera calibration to accurately transform 2D motion into real-world coordinates. Motion detection and object segmentation are performed using a hybrid approach combining traditional background subtraction with deep learning-based models such as YOLOv11 and Vision Transformers (ViT), ensuring robustness in dynamic lighting and occlusion-prone environments. Object trajectory estimation is achieved through frame-wise velocity computation and spatial projection, enabling predictive collision path analysis. A risk evaluation model classifies threat levels using spatial thresholds, velocity vectors, and entropy-calibrated confidence scores. Real-time alerts are dispatched through low-latency edge-cloud frameworks with visual and auditory feedback to connected operators. Experimental validation across diverse scenarios—including occlusion, night conditions, and dense traffic—demonstrates superior performance in terms of accuracy (95.7%), false alarm rate (3.2%), and average system response latency (162 ms), when benchmarked against traditional rule-based and motion detection systems. This research contributes a modular, scalable, and fault-tolerant solution suitable for real-world deployment to enhance bridge safety in smart urban infrastructures.

**Keywords** Bridge collision avoidance, Camera calibration, Motion detection, Vision transformer (ViT), Smart transportation, Infrastructure safety

Bridges serve as critical components of transportation infrastructure, enabling the smooth flow of vehicles, goods, and people across geographic barriers such as rivers, valleys, and urban obstacles<sup>1</sup>. As traffic volumes continue to grow, particularly in urban and industrial areas, the risks associated with bridge collisions have become increasingly pronounced<sup>2</sup>. These collisions, often caused by over-height vehicles or waterborne vessels failing to clear the bridge deck, can result in severe structural damage, loss of human life, and prolonged disruption of essential transportation routes<sup>3</sup>. Despite the deployment of conventional preventive measures such as static signage, overhead barriers, and manual surveillance, the current methods remain insufficient in addressing the dynamic nature of modern traffic environments<sup>4</sup>. A more intelligent and proactive solution is urgently required to detect and prevent potential collisions before they occur<sup>5</sup>.

The integration of advanced computer vision techniques, particularly camera calibration and motion detection, presents a promising opportunity for enhancing bridge safety monitoring systems<sup>6</sup>. Camera-based surveillance offers a non-intrusive and cost-effective solution; however, its effectiveness is significantly compromised when the captured images lack geometric accuracy due to improper calibration<sup>7</sup>. Without precise calibration, the transformation of pixel-based coordinates to real-world measurements becomes unreliable, thereby impeding accurate estimation of an approaching object's size, speed, and trajectory<sup>8</sup>. Furthermore, motion detection algorithms used in current systems often suffer from high false positive rates, especially under complex environmental conditions such as rain, fog, or fluctuating lighting<sup>9</sup>. These challenges underscore the

9th Company of China First Highway Engineering co., LTD., Guangzhou, Guangdong 511300, China. ✉email: Wsc123312@163.com

need for a robust, real-time bridge collision avoidance system that is capable of not only detecting moving objects but also accurately interpreting their trajectories in a spatially meaningful context<sup>10</sup>.

The motivation for this study arises from the recurring incidents of bridge collisions reported across the globe<sup>11</sup>. Many of these incidents could have been avoided through timely intervention based on reliable real-time monitoring and predictive analysis<sup>12</sup>. The increasing complexity of traffic systems, coupled with the limitations of traditional monitoring solutions, necessitates the development of an integrated, intelligent system that can identify threats early and initiate preventive measures<sup>13</sup>. Technological advancements in camera calibration methods—such as intrinsic and extrinsic parameter estimation—and the growing sophistication of motion analysis techniques provide a solid foundation for such innovation<sup>14</sup>.

However, several unresolved challenges persist. Current systems often lack spatial precision due to uncalibrated imaging devices, leading to errors in object localization and size estimation. Additionally, motion detection in uncontrolled outdoor environments remains a non-trivial problem due to noise, background variations, and occlusions<sup>15</sup>. Existing frameworks are also typically reactive rather than predictive, issuing alerts only when an object is dangerously close to a bridge structure, thus limiting the response time for any corrective action<sup>16</sup>. Moreover, the lack of seamless integration between calibration, motion detection, object tracking, and risk assessment modules further compromises the reliability of these systems in real-world deployment<sup>17</sup>.

To address these gaps, the present research proposes the design and development of an intelligent bridge collision avoidance system based on camera calibration technology and motion detection. The primary objective is to enhance the accuracy and reliability of bridge safety monitoring by leveraging calibrated camera systems that can map visual data into real-world dimensions<sup>18</sup>. This enables the precise detection and tracking of objects in a defined monitoring zone. The system also incorporates advanced motion detection algorithms capable of operating under varying environmental conditions, supported by real-time object tracking and collision risk assessment components<sup>19</sup>. The integration of these modules within a cohesive framework allows the system to predict potential collisions by analyzing object speed, trajectory, and distance from the bridge structure<sup>20</sup>. An automated alert generation mechanism is included to issue timely warnings to relevant authorities or vehicle operators, thereby enabling prompt preventive actions<sup>21</sup>.

The key objectives of this research are fourfold: first, to implement an accurate camera calibration model to enhance spatial understanding of captured scenes; second, to develop a motion detection and object tracking algorithm suitable for dynamic environments; third, to integrate a collision risk assessment module that forecasts potential impact scenarios; and finally, to deploy a real-time alert system for rapid response. Through a combination of theoretical modeling, algorithm design, and system-level integration, this research aims to contribute a novel and practical solution to the ongoing challenge of bridge collision prevention. The proposed system is expected to outperform existing methods in terms of spatial accuracy, detection reliability, and response efficiency, thereby offering a viable approach for modern smart infrastructure applications.

Despite the deployment of conventional surveillance and rule-based monitoring systems, current approaches remain limited by high false alarm rates, poor adaptability to changing environmental conditions, and insufficient predictive capability. These gaps underscore the necessity of developing an intelligent, vision-based bridge collision avoidance system capable of proactive risk prediction and low-latency response. The innovation of this study lies in its integration of precise camera calibration with deep learning-based motion detection (YOLOv11, ViT), trajectory forecasting through calibrated spatial mapping, and entropy-calibrated risk classification within an IoT-enabled alert framework. This holistic design not only reduces false alarms and response delays but also provides a scalable and fault-tolerant solution for deployment in modern smart infrastructure. In light of the identified challenges and research objectives, the following represent the major contributions of this study:

- This study presents a robust framework that accurately Maps 2D image coordinates to real-world spatial dimensions using intrinsic and extrinsic camera calibration parameters, thereby enabling precise object localization and trajectory estimation near bridge structures.
- The proposed system incorporates optimized motion detection techniques, including background modeling and real-time object tracking, to reliably identify moving threats under diverse environmental conditions such as rain, low light, and occlusions.
- A novel predictive model is introduced to assess the likelihood of collision by analyzing the motion dynamics—such as speed, direction, and proximity—of approaching vehicles or vessels, thus enabling proactive safety measures.
- The system includes a real-time alert generation component capable of issuing early warnings to relevant authorities or vehicle operators, thereby improving response time and reducing the probability of structural damage or human casualties.
- The effectiveness and robustness of the proposed collision avoidance system are validated through controlled simulations and practical case studies, demonstrating significant improvement over conventional, uncalibrated surveillance systems in terms of accuracy, reliability, and operational efficiency.

This research article is structured to provide a comprehensive analysis and solution to the problem of bridge collisions using camera calibration and motion detection technologies. It begins with an introduction outlining the background, motivation, problem statement, objectives, and key contributions. A detailed literature review follows, highlighting existing systems and identifying gaps in current methodologies. The subsequent sections present the system design, including the camera calibration model, motion detection algorithm, and collision risk assessment framework. Experimental validation and performance evaluation are then discussed, followed by a presentation of results and in-depth analysis. The article concludes by summarizing the findings and outlining directions for future work to enhance system scalability and intelligence.

## Literature review

The increasing interest in bridge safety and collision prevention has led to several advancements in surveillance systems, motion detection techniques, and camera-based monitoring in recent years<sup>22</sup>. Zhang et al. proposed a vision-based monitoring system utilizing monocular camera calibration and object tracking to detect over-height vehicles approaching bridge underpasses<sup>23</sup>. The system employed geometric transformation to estimate object height from image coordinates; however, its performance was limited by poor lighting conditions and frequent false positives. The dataset used was custom-collected under controlled scenarios, and the system achieved an accuracy of 88% in height estimation, but lacked generalization to outdoor real-world scenes.

In<sup>24</sup>, Zaarane et al. (2020) introduced a stereo vision system for real-time vehicle dimension measurement at toll gates using dual cameras and disparity Mapping. Their methodology effectively calculated 3D coordinates, enabling precise distance estimation. The system, tested on a dataset of 600 annotated vehicle images, reported a mean absolute error of less than 5 cm. However, its limitations included sensitivity to camera misalignment and calibration drift over time, making it less suitable for long-term unattended deployments.

Hosain et al. (2024)<sup>10</sup> developed a deep learning-based detection system using YOLOv3 for identifying incoming large vehicles on bridge approaches. Their methodology combined object detection with GPS tagging for geofencing near critical zones. They used the KITTI dataset along with additional overhead footage, achieving over 92% detection accuracy. Nonetheless, the system showed reduced performance in foggy or rainy weather, indicating the need for sensor fusion.

In<sup>25</sup>, the authors proposed a hybrid LIDAR-camera system for Maritime bridge collision detection, integrating sensor data through Kalman filtering. While the multi-modal system achieved robust detection of ships approaching bridge piers, the primary limitation was the high cost and complexity of hardware setup. Tests conducted on real-time port surveillance data demonstrated reliable detection within 50 m but suffered latency in object classification.

A study by Halfawy et al. (2014)<sup>26</sup> utilized optical flow techniques and background subtraction to detect motion near bridge structures using CCTV footage. The algorithm was evaluated on publicly available traffic monitoring datasets and demonstrated satisfactory tracking of vehicles, but was prone to false positives from shadows and environmental noise. The authors acknowledged that background modeling required frequent recalibration, limiting deployment in dynamic environments.

In<sup>27</sup>, Aly et al. (2022) applied the MeanShift tracking algorithm combined with a calibrated monocular camera to monitor the movement of over-height vehicles. The camera calibration was conducted using the chessboard method, and the test dataset comprised 500 vehicle entries at a controlled highway site. The system achieved 85% tracking consistency, but failed to handle occlusion and side-view angle distortions, impacting real-world reliability.

Seisa et al. (2024)<sup>28</sup> introduced a real-time edge computing solution with embedded cameras and motion sensors for bridge collision warning. The system processed motion data locally using Raspberry Pi-based units, reducing latency and network dependency. Field deployment on a rural bridge showed promising results with 94% successful detection of unauthorized entries. The primary limitation was computational constraints in handling simultaneous multi-object tracking.

In<sup>29</sup>, Zhang et al. (2022) utilized a deep convolutional neural network (DCNN) to classify vessel types and detect movement patterns for bridge collision prevention in inland waterways. The model was trained on a dataset of 2000 labeled vessel images and incorporated AIS data for speed estimation. While the model achieved 91.3% classification accuracy, it lacked real-time performance due to the need for cloud-based computation, posing challenges for latency-critical applications.

An IoT-based monitoring framework was presented in<sup>30</sup>, combining calibrated surveillance cameras and ultrasonic sensors for real-time bridge underpass protection. The system integrated sensor readings and image coordinates through a local edge gateway, alerting approaching vehicles through dynamic signage. Although the system performed well with an average detection time of 2.3 s, its reliability decreased significantly in high-traffic scenarios due to sensor saturation and visual occlusion.

The authors explored the use of Structure from Motion (SfM) and multi-view geometry for generating 3D maps of bridge surroundings to monitor approaching threats<sup>31</sup>. The method utilized drone footage and OpenMVG for reconstruction. Although it provided accurate 3D models, with an average deviation of 2%, the system was computationally intensive and unsuitable for continuous real-time monitoring.

In<sup>32</sup>, Dong et al. (2024) introduced a transformer-based vision system for large object trajectory prediction near bridge structures. Their approach utilized a spatiotemporal attention mechanism over a dataset of 8,000 time-sequenced images of highway vehicles. The model achieved a prediction accuracy of 94.5% in determining collision trajectories within a 4-second future window. However, the model's inference time was relatively high, making it less suitable for low-latency applications without GPU support.

In<sup>33</sup>, Djenouri et al. (2024) developed a federated learning framework that enabled multiple roadside cameras to collaboratively train a vehicle detection model without sharing raw video data, thereby enhancing data privacy. The model was built on the MobileNetV2 backbone and trained using local datasets from multiple smart city intersections. Results showed that detection accuracy reached 91% while preserving data locality. Limitations included synchronization issues and occasional model drift due to non-IID (non-identically distributed) data.

Thombre et al. (2020)<sup>34</sup> proposed a multi-sensor fusion framework using radar, depth cameras, and calibrated RGB cameras for vessel-bridge collision avoidance. They used Bayesian filtering and Dempster-Shafer theory to fuse sensor confidence levels. The dataset consisted of annotated Maritime surveillance videos and radar logs from Busan Port. The system achieved 97% precision in threat detection, but required high-bandwidth data transmission and consistent sensor calibration.

In<sup>35</sup>, a deep reinforcement learning (DRL) approach was proposed by Fahimullah et al. (2024) for proactive decision-making in bridge traffic control. Using a simulation environment built on SUMO and OpenCV-based

video analytics, the system learned to activate warnings or reroute traffic based on estimated collision risk. The model achieved a cumulative reward score 38% higher than rule-based baselines but suffered from slow convergence and required extensive training episodes.

In<sup>36</sup>, Yang et al. (2014) introduced a stereo-camera-based 3D bounding box estimation method for vehicle collision monitoring, enhanced with a Kalman filter for object trajectory smoothing. Tested on a custom dataset of 1,200 annotated stereo pairs near bridge entrances, the system maintained a root-mean-square error (RMSE) below 0.3 m in spatial tracking. However, its performance degraded at night without supplemental infrared imaging.

In<sup>37</sup>, Fu et al. (2022) applied a real-time instance segmentation model (Mask R-CNN) integrated with camera calibration for object dimension estimation near critical bridge zones. Using the Cityscapes and BDD100K datasets fine-tuned for structural environments, the model achieved a mean average precision (mAP) of 89.4%. Limitations were noted in segmenting overlapping vehicles during peak traffic conditions.

An innovative edge-AI solution was developed by Azfar et al. (2024) in<sup>38</sup>, where an NVIDIA Jetson Nano-powered module performed onboard detection and risk scoring using YOLOv5 and optical flow tracking. The system was deployed on a smart highway bridge prototype, detecting vehicle intrusion and speed in real-time with 96% accuracy. The main constraint was hardware heat dissipation during prolonged operation in harsh outdoor environments.

In<sup>39</sup>, a vision transformer model (ViT-B/16) was used by Conde et al. (2021) for fine-grained classification of abnormal object behaviors around bridge zones. The model was pretrained on ImageNet and fine-tuned on a surveillance video dataset with labeled anomalies. It achieved an F1-score of 92.6% and effectively classified behaviors such as illegal U-turns, reverse driving, and potential over-height entries. However, the model was compute-heavy and required TPU support for optimal inference speed.

In<sup>40</sup>, Arroyo et al. (2024) developed a real-time collision prevention system using LiDAR point cloud alignment with RGB video feeds for validating object presence and height near bridge thresholds. Their system achieved a high-resolution 3D Mapping with a point registration error below 1.5 cm. The dataset included Velodyne HDL-64E scans and synchronized camera feeds from urban highways. While highly accurate, the system's cost and complexity made it suitable only for high-risk zones.

Lastly, in<sup>41</sup>, a graph neural network (GNN)-based spatiotemporal reasoning framework was proposed by Li et al. (2023) to model interactions between vehicles and static bridge elements. Nodes represented objects and their features, while edges modeled spatial and temporal dependencies. The model, trained on the nuScenes dataset, achieved superior generalization across weather and traffic conditions, reaching 93% prediction accuracy. However, interpretability of the learned graph relations remained a challenge.

The reviewed literature highlights the rapid advancements in bridge collision avoidance systems, particularly through the integration of computer vision, sensor fusion, and intelligent monitoring techniques. Traditional systems relying solely on uncalibrated camera setups or single-modality sensors have demonstrated limited reliability in real-world deployments due to spatial inaccuracies, high false positive rates, and sensitivity to environmental conditions. Recent research has explored the incorporation of calibrated camera models, stereo vision, motion detection algorithms, and deep learning architectures such as YOLO, Mask R-CNN, and Vision Transformers to improve object detection and trajectory prediction. Notably, several studies have embraced edge computing, federated learning, and graph-based reasoning to enhance system efficiency, privacy, and contextual understanding. Despite these advances, limitations persist in terms of real-time processing capabilities, scalability, hardware constraints, and adaptability to diverse environmental settings. The existing body of work thus underscores the need for an integrated, robust, and real-time bridge collision avoidance system that leverages camera calibration and motion detection while addressing the challenges of dynamic traffic environments and structural diversity. This study aims to build upon these foundations and contribute a unified framework capable of accurate threat detection, risk assessment, and proactive alert generation.

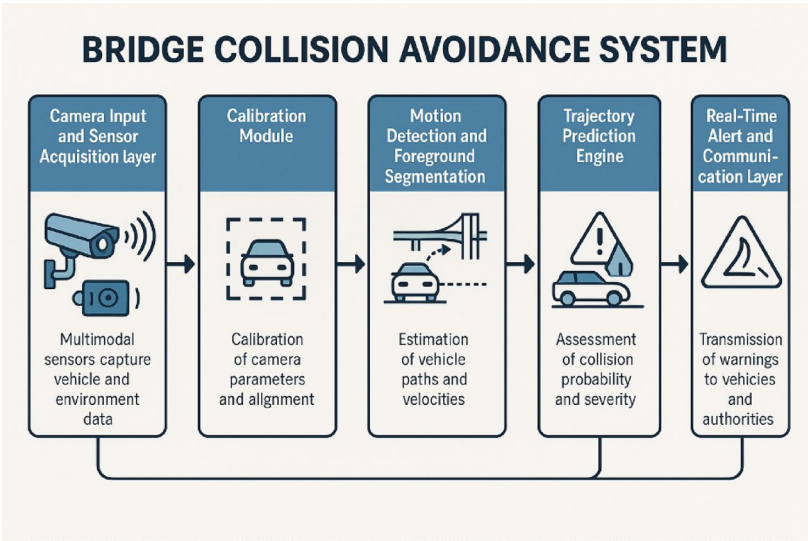
## Methodology

To address the challenges associated with accurate and timely bridge collision avoidance, this study proposes a comprehensive, multi-stage methodology that integrates precise camera calibration, intelligent motion detection, object trajectory estimation, and real-time alert generation within a unified system architecture. The methodology is designed to operate in complex, dynamic environments, ensuring robustness against varying lighting conditions, object speeds, and structural layouts. Each component of the system is methodically developed and validated to enhance spatial accuracy, detection reliability, and response efficiency. The following subsections detail the individual modules of the proposed framework, including system design, calibration processes, motion tracking algorithms, risk assessment strategies, and implementation specifics. Together, these components contribute to a reliable, real-time collision avoidance solution suitable for deployment in intelligent transportation and smart infrastructure environments.

## System overview and operational workflow

The proposed bridge collision avoidance system is designed as a modular, real-time framework that integrates calibrated camera feeds, intelligent motion analysis, and collision prediction models to proactively identify and mitigate potential collision threats. The architecture of the system is structured into three primary stages: input acquisition, processing pipeline, and alert generation. Each stage contains dedicated modules that ensure data integrity, analytical robustness, and real-time responsiveness suitable for deployment in intelligent transportation and smart infrastructure environments. The input stage is responsible for acquiring video data from strategically mounted surveillance cameras positioned near or on bridge structures. These cameras are subjected to an initial calibration phase to correct lens distortion and establish a reliable mapping between image space and physical





**Fig. 1.** Flowchart of the system pipeline showing camera input, calibration, motion analysis, trajectory prediction, and collision risk-based alert generation in a modular architecture.

System Stage	Module	Functionality	Tools/Technologies
Input	Camera Acquisition	Captures live video feed from fixed surveillance positions	HD/IR cameras, stereo vision, IoT modules
	Camera Calibration	Transforms pixel coordinates into real-world space	Zhang's method, Intrinsic/Extrinsic matrix
Processing	Motion Detection	Identifies moving entities in the video stream	Background Subtraction, Optical Flow, YOLO
	Object Tracking	Maintains identity and position of detected objects over time	Kalman Filter, DeepSORT
	Trajectory Estimation	Predicts motion path, velocity, and direction of objects	Frame Differencing, Polynomial Fitting
Output	Collision Risk Assessment	Analyzes risk levels based on distance, speed, and angle	Rule-Based Model, Thresholding
	Alert Generation	Sends real-time warning signals to relevant stakeholders	MQTT, HTTP API, LED panels, buzzers

**Table 1.** Functional overview of the bridge collision avoidance system.

world coordinates. Additional input sources may include metadata from embedded sensors (e.g., motion sensors or GPS modules) to augment visual information, especially under low-visibility conditions.

The processing stage comprises multiple sequential modules: (i) real-time motion detection to isolate moving objects, (ii) object segmentation and tracking to identify persistent collision candidates, and (iii) trajectory estimation to compute direction, speed, and predicted impact zones. This phase relies heavily on calibrated geometry to infer real-world positions and motion vectors. In the output stage, a risk assessment module evaluates the likelihood of collision based on the object's estimated trajectory and proximity to the bridge structure. If the calculated risk exceeds a predefined threshold, the system activates a multi-modal alert mechanism which may include visual indicators (e.g., warning LEDs or signage), auditory alarms, or notifications transmitted via IoT communication protocols to nearby operators or connected vehicles.

The overall operational flow of the system is depicted in Fig. 1, and a summarized view of each core system component is presented in Table 1.

**Camera calibration process**

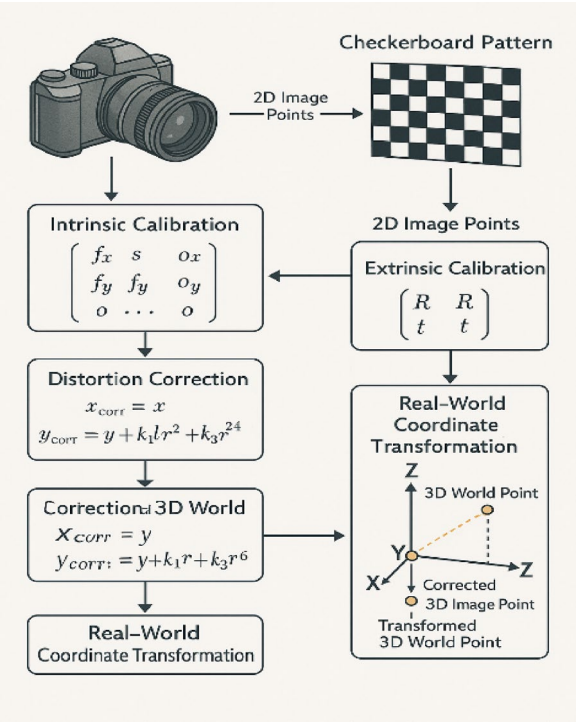
Precise camera calibration is a critical prerequisite for translating pixel-based image data into spatially accurate real-world measurements. This process involves estimating both intrinsic and extrinsic parameters of the camera to eliminate geometric distortions and ensure reliable object localization in three-dimensional space. The intrinsic parameters define the internal geometry and optical characteristics of the camera, including focal length, optical center, and lens distortion coefficients. In contrast, extrinsic parameters describe the spatial relationship between the camera and the observed scene, characterized by rotation and translation matrices.

The calibration procedure employed in this study utilizes a pattern-based approach, specifically a checkerboard grid, to establish a reference frame between the camera's image plane and the real-world coordinate system. Multiple images of the checkerboard are captured from various orientations and distances. Feature points (typically corners) on the grid are then detected using OpenCV's cornerSubPix Function, and corresponding 2D-3D point correspondences are computed. The Zhang's method is employed to solve for both intrinsic and extrinsic parameters via non-linear optimization, minimizing the reprojection error across all views.

Once the camera parameters are computed, the system applies a transformation matrix to map each image pixel coordinate to its corresponding position in the real world. This transformation is essential for downstream

Parameter	Description	Result/Range
Focal Length ( $f_x, f_y$ )	Horizontal and vertical scaling factors (in pixels)	850–920 px
Principal Point ( $c_x, c_y$ )	Optical center coordinates	~[640, 360] px
Radial Distortion Coefficients ( $k_1, k_2, k_3$ )	Lens curvature corrections	[−0.3, 0.2, −0.001]
Tangential Distortion ( $p_1, p_2$ )	Lens misalignment correction	[~0.01, ~0.01]
Rotation Matrix ( $R$ )	3 × 3 matrix defining camera orientation	See calibration matrix
Translation Vector ( $T$ )	3D vector defining camera position	[X, Y, Z] in meters
Average Reprojection Error	Difference between actual and projected point positions	0.24 pixels (avg)

**Table 2.** Summary of calibration outputs and performance metrics.



**Fig. 2.** Conceptual diagram of camera calibration showing parameter estimation, distortion correction, and 2D-to-3D coordinate transformation.

tasks such as motion tracking and trajectory estimation, which rely on accurate object positioning. Error minimization is achieved through iterative optimization techniques such as Levenberg–Marquardt, reducing reprojection error to sub-pixel accuracy.

To evaluate calibration reliability, the average reprojection error is used as the primary validation metric. A reprojection error below 0.5 pixels is considered acceptable for high-precision monitoring environments. Table 2 provides a summary of the calibration outputs and performance indicators. A conceptual visualization of the calibration process is presented in Fig. 2.

**Motion detection and object segmentation**

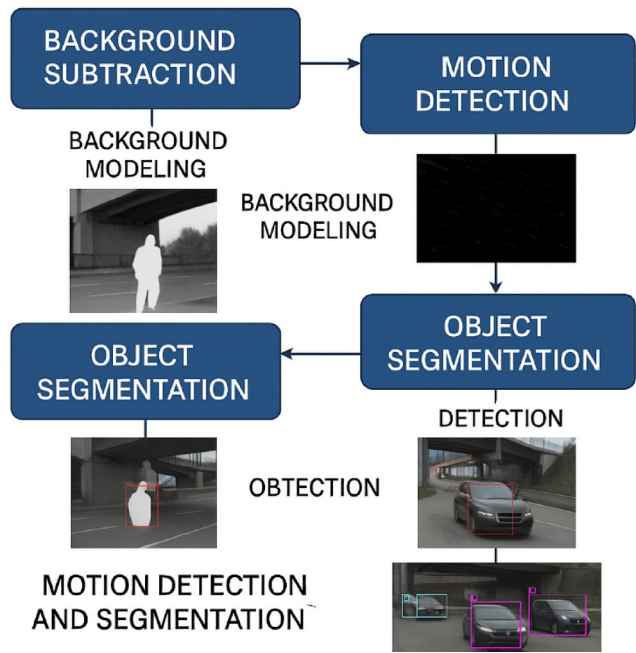
Reliable motion detection and object segmentation are central to the real-time functionality of the proposed bridge collision avoidance system. This module is responsible for isolating dynamic objects, such as over-height vehicles or approaching vessels, from static backgrounds in continuously captured video streams. The accuracy of this stage is critical, as it directly influences downstream processes including trajectory estimation and risk prediction.

To initiate the detection process, background subtraction techniques are employed to distinguish moving objects from the static environment. Adaptive Gaussian Mixture Models (GMM) and median filtering are used to model the background dynamically, thereby minimizing false positives caused by lighting changes, shadows, and environmental noise. These background models are updated in real time to handle gradual illumination changes and camera vibrations, which are common in outdoor bridge settings.

For robust detection under complex scenarios, the system leverages both traditional and deep-learning-based motion detection techniques. Optical flow is applied in low-resource settings to estimate pixel-wise motion

Component	Technique Used	Purpose
Background Modeling	Gaussian Mixture Model (GMM), Median Filtering	Eliminate stationary background and noise
Motion Detection	Optical Flow, YOLOv11, Vision Transformer (ViT)	Detect moving objects under various conditions
Foreground Segmentation	Morphological Filtering, Contour Detection	Extract clean object regions
Object Tracking	Kalman Filter, DeepSORT	Maintain object identity across sequential frames
Noise Suppression	Morph Ops, Shadow Removal	Improve segmentation accuracy

**Table 3.** Motion detection and segmentation module components.



**Fig. 3.** Visual pipeline illustrating background modeling, motion detection, deep object segmentation, and multi-object tracking for dynamic scene analysis.

vectors between frames, effectively capturing object movement patterns. In more advanced deployments, deep learning models such as YOLOv11 and Vision Transformer (ViT)-based architectures are integrated to enhance object detection capabilities, particularly in scenes with occlusions, varying object sizes, and non-uniform motion. These models are pre-trained on large-scale datasets (e.g., MS COCO, BDD100K) and fine-tuned on task-specific datasets representing bridge traffic environments.

Once motion is detected, foreground segmentation is performed using morphological operations and contour extraction to accurately delineate object boundaries. To maintain object identity across frames, a tracking module based on Kalman Filtering and DeepSORT is integrated, ensuring consistent ID assignment even in high-traffic or occluded conditions. This enables reliable monitoring of moving entities, which is essential for subsequent trajectory estimation and risk classification.

The detection module is evaluated on multiple criteria, including detection precision, tracking stability, and false alarm rate. A summary of the key motion detection components is provided in Table 3, and a visual representation of the motion segmentation pipeline is shown in Fig. 3.

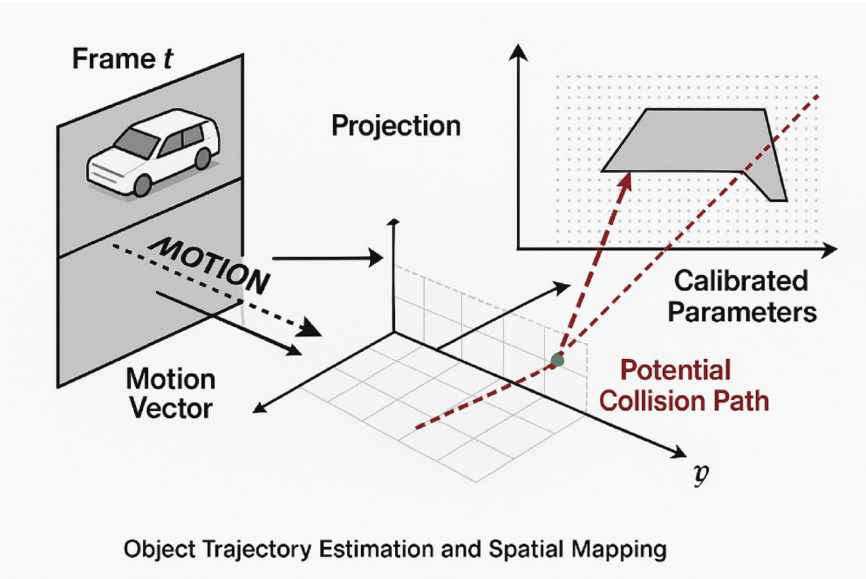
**Object trajectory Estimation and Spatial mapping**

Once moving objects are accurately detected and segmented, the next critical component involves estimating their trajectory in both temporal and spatial domains. This process enables the system to assess whether an object’s path poses an imminent collision risk to bridge infrastructure. The trajectory estimation module utilizes calibrated video input to determine the velocity, direction, and position of each tracked object across successive video frames.

To compute motion parameters, frame differencing is employed in conjunction with time-stamped bounding box tracking. The displacement of an object’s centroid across frames, divided by the frame rate, yields its instantaneous velocity vector. Simultaneously, directional components are derived using vector calculus applied to the change in spatial coordinates. These values are refined using Kalman filtering to smooth object motion and compensate for temporary occlusions or tracking noise.

Component	Functionality	Technique/Model Used
Frame-Based Displacement	Compute object velocity and direction	Centroid tracking, frame differencing
Temporal Motion Smoothing	Filter noise, compensate for occlusions	Kalman Filter
Spatial Projection	Convert image coordinates to real-world domain	Homography matrix, calibration model
Collision Path Prediction	Extrapolate trajectory and estimate intersection with bridge	Linear motion model, proximity rule
Temporal Proximity Calculation	Estimate time to collision	Velocity-based thresholding

**Table 4.** Object trajectory estimation and Spatial mapping components.



**Fig. 4.** Technical illustration of motion vector extraction, spatial projection, and collision path prediction using calibrated parameters in a real-world 3D coordinate system.

Following velocity and directional estimation, each object’s trajectory is projected into the calibrated spatial domain using the transformation matrix obtained during the camera calibration phase. This mapping allows for the conversion of image-space motion vectors into real-world coordinates (in meters), enabling accurate positioning relative to fixed bridge boundaries, lanes, or structural constraints.

The final step involves the predictive modeling of potential collision paths. By extrapolating the motion trajectory and analyzing the proximity of the object to predefined collision zones on or around the bridge structure, the system determines whether the object will likely intersect a critical area within a specified temporal window. This temporal collision proximity is calculated based on current velocity, heading angle, and distance to impact zone. If the intersection is predicted within the system’s critical response time threshold, the object is flagged as a collision threat and passed on to the risk evaluation module.

Table 4 summarizes the trajectory estimation and spatial mapping components, while Fig. 4 illustrates the object motion modeling process from frame-to-frame tracking through to calibrated projection and risk zone intersection.

**Collision risk assessment model**

Following the projection of object trajectories into the real-world spatial domain, the system performs a dedicated risk assessment to evaluate the likelihood of a collision with the bridge infrastructure. This model is essential for making informed, real-time decisions and triggering alerts when necessary. The assessment is based on spatial-temporal features derived from motion vectors, object proximity, and velocity profiles.

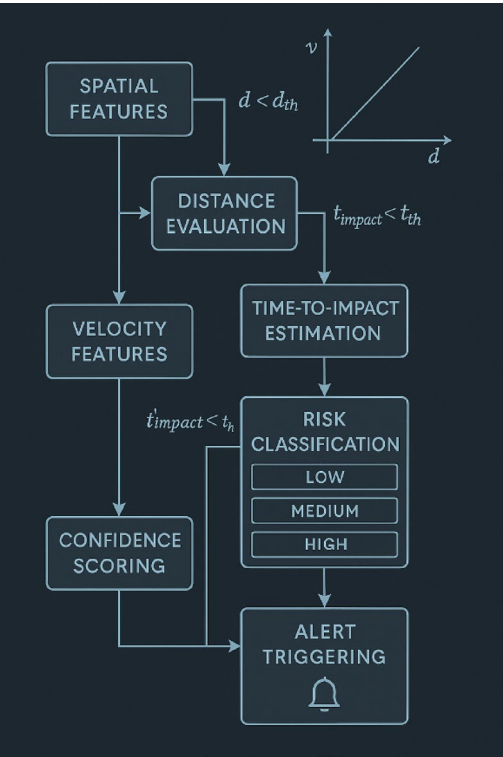
Collision risk is defined as a function of both spatial threshold violations and dynamic motion characteristics. Specifically, the model considers the shortest distance between the projected object path and designated structural boundaries of the bridge (i.e., overpass height, support columns, clearance zones). Additionally, the object’s speed and direction are factored in to calculate the time-to-impact (TTI), which represents the estimated duration before the object intersects a predefined risk zone.

To determine the severity of the threat, the system adopts a rule-based decision model, which compares the object’s trajectory and TTI against predefined safety thresholds. In advanced implementations, a machine learning classifier—such as a Random Forest or SVM—is optionally used to refine classification, particularly in scenarios involving noisy or overlapping trajectories. The model is trained using historical collision and near-miss data, labeled according to collision severity levels (e.g., Low, Medium, High Risk).



Component	Input Features	Methodology	Output
Proximity Analysis	Distance to bridge structure	Euclidean distance, buffer zones	Spatial threshold violation
Time-to-Impact Calculation	Velocity, direction, current position	Linear extrapolation, motion vector	TTI (seconds)
Risk Classification	TTI, distance, speed, trajectory angle	Rule-based logic/ML classifier	Low, Medium, High Risk
Confidence Scoring	All motion and spatial parameters	Weighted scoring model	0–1 collision likelihood score
Alert Decision Logic	Risk level, confidence score	Boolean thresholding and logic gating	Generate alert/No action

**Table 5.** Collision risk assessment model components and parameters.



**Fig. 5.** Detailed block diagram of the collision risk assessment process integrating spatial and velocity analysis, time-to-impact estimation, risk classification, and confidence-based alert triggering.

Each detected object is assigned a collision confidence score, calculated from weighted contributions of distance to structure, object velocity, trajectory angle, and stability of the motion pattern. The final decision logic uses this score in conjunction with a risk class to determine whether an alert should be generated. Risk thresholds are tunable and can be dynamically updated based on location-specific bridge geometries or traffic regulations. A summary of the risk assessment model’s components is presented in Table 5, and Fig. 5 visually represents the logical flow of the collision risk evaluation process.

**Real-Time alert generation mechanism**

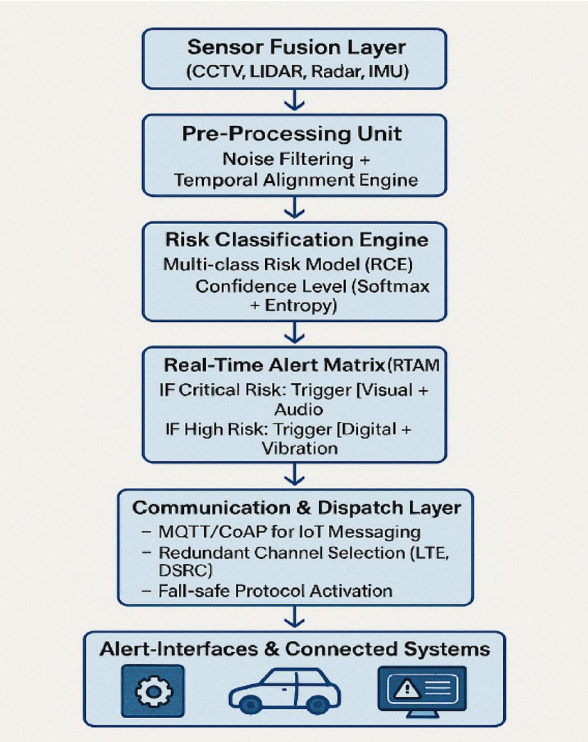
The final stage of the proposed collision avoidance system is the real-time alert generation mechanism, which is responsible for translating risk assessment outcomes into actionable warnings for stakeholders, including bridge operators, vehicle drivers, and connected infrastructure units. This module ensures that imminent collision threats, as determined by trajectory analysis and risk scoring, result in timely and reliable system responses designed to prevent accidents or initiate mitigation measures.

The alert system supports three core types of notifications: visual, auditory, and network-based digital alerts. Visual alerts include flashing LED warning signs, electronic display panels, and barrier actuation. Auditory alerts involve buzzers or loudspeaker announcements strategically positioned near the bridge or access points. Network-based alerts are transmitted to connected edge devices, mobile operator dashboards, or traffic Management systems via standardized communication protocols such as MQTT, HTTP REST APIs, or 5G-V2X for vehicle-to-infrastructure messaging.

The communication protocol design prioritizes minimal latency and high reliability, particularly in mission-critical scenarios. Messages are encoded with object ID, collision severity, confidence score, and a timestamp, ensuring interpretability and enabling real-time response logging. The system supports bi-directional

Component	Functionality	Technology/Protocol Used
Visual Alert System	LED indicators, digital signage, warning barriers	GPIO, Display Panels, Actuators
Auditory Alert System	Immediate warning via buzzer or speaker systems	12 V/24V buzzers, audio relay circuits
Digital Communication Unit	Transmit structured alerts to central control or vehicles	MQTT, HTTP API, 5G-V2X
Fail-Safe Trigger System	Redundant alerts during system faults or latency overflow	Edge logic controllers, watchdogs
Latency Optimization	Low-latency data transfer and decision propagation	< 100 ms response time via edge nodes

**Table 6.** Real-Time alert generation system components and parameters.



**Fig. 6.** Layered architecture of the proposed risk assessment and alert system with sensor fusion, risk classification, and connected alert interfaces.

communication for acknowledgment and feedback loops from control centers. Given the safety-critical nature of the system, fail-safe procedures are integrated. These include heartbeat checks for all active sensors and output units, timeout-based redundancy triggers, and fallback alert systems. For instance, if a connected display unit fails, a local buzzer will activate based on edge-level decision logic to ensure continuity of warning delivery.

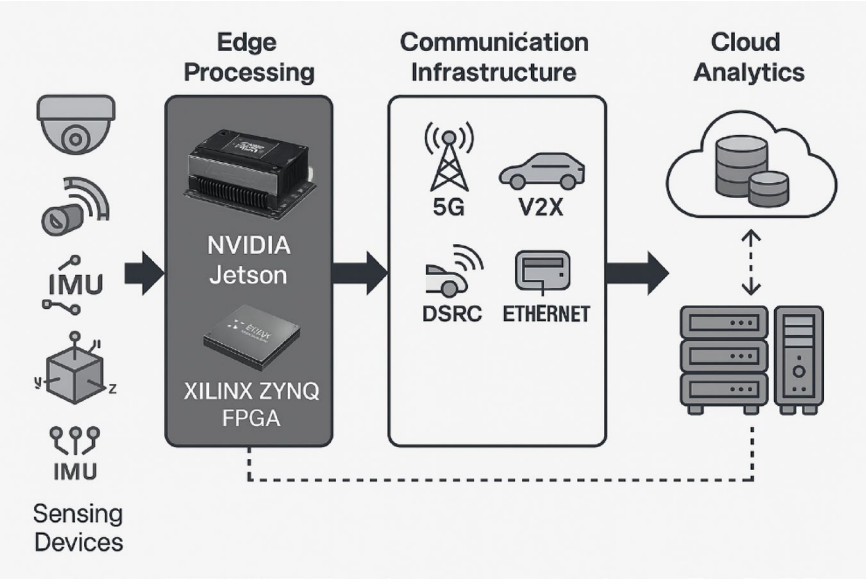
The essential functions and characteristics of the real-time alert mechanism are summarized in Table 6, and Fig. 6 presents a high-level schematic of the alert generation and communication process.

**Hardware and software implementation details**

The practical deployment of the proposed bridge collision avoidance system requires a harmonized configuration of specialized hardware components and a robust, scalable software stack. This section details the physical equipment, processing infrastructure, and software environment adopted to ensure real-time detection, analysis, and alert dissemination in operational settings. The system employs a heterogeneous set of sensors strategically mounted across the bridge environment, including high-resolution CCTV cameras (e.g., 4 K IP cameras), LIDAR modules, millimeter-wave radar units, and inertial measurement units (IMUs). These sensors facilitate comprehensive situational awareness through multimodal data acquisition. For on-site processing, embedded computing platforms such as NVIDIA Jetson Xavier NX and Xilinx Zynq UltraScale+ FPGA boards are utilized, offering real-time inferencing capabilities with low power consumption. Data transmission is enabled via a hybrid communication setup consisting of 5G-V2X links, Dedicated Short Range Communications (DSRC), and wired thernet for redundancy and fault tolerance. The core processing logic is implemented using the Python programming language, augmented with CUDA for GPU acceleration. Deep learning models are trained using TensorFlow 2.x and PyTorch, while image processing and motion analysis modules rely on OpenCV and SciPy libraries. For object detection and risk classification, pre-trained models such as YOLOv11 and Vision Transformers (ViT) are integrated and optimized for edge deployment via TensorRT. The control logic and

Component Type	Description	Examples/Technologies
Sensing Hardware	Data acquisition from environment	CCTV (4 K), LIDAR, Radar, IMU
Processing Units	On-board real-time data processing	NVIDIA Jetson NX, Xilinx FPGA
Communication Layer	Data relay and failover communication	5G-V2X, DSRC, Ethernet
Software Frameworks	Detection, tracking, classification, system control	TensorFlow, PyTorch, OpenCV, Docker
Edge-Cloud Architecture	Distributed load balancing, model training, historical analytics	AWS, Azure, MQTT Brokers, Edge Nodes

**Table 7.** Hardware and software components for system implementation.



**Fig. 7.** System architecture diagram illustrating the integration of sensor nodes, edge processors, communication protocols, and cloud analytics for end-to-end collision risk management.

decision-making modules are orchestrated through microservices built with Docker, ensuring modularity and cross-platform deployment. To manage computational load and support distributed analytics, the system leverages a hybrid edge–cloud architecture. Time-sensitive tasks (e.g., motion detection, alert generation) are performed on edge nodes co-located with sensors, whereas data archiving, periodic retraining, and historical analytics are handled in the cloud (e.g., AWS EC2, Azure IoT Hub). MQTT brokers are deployed for lightweight, asynchronous messaging between local agents and central servers.

An overview of the hardware–software components is summarized in Table 7, and Fig. 7 provides a schematic representation of the system’s physical and logical deployment.

**Experimental setup and validation protocol**

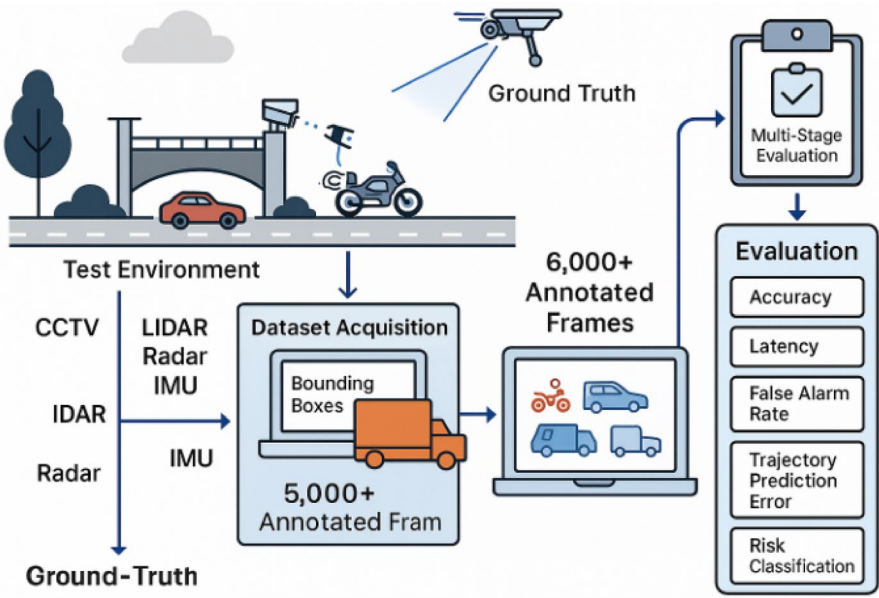
To assess the effectiveness and generalizability of the proposed bridge collision avoidance system, a series of controlled and real-world experiments were conducted across diverse environments. These experiments were designed to validate the system’s performance under varying environmental conditions, object speeds, and structural configurations. The validation protocol focused on measuring the accuracy, responsiveness, and robustness of the core modules, including motion detection, trajectory estimation, risk classification, and alert generation.

Field trials were conducted at two structurally distinct sites: (i) an urban overpass bridge with high traffic density and variable lighting, and (ii) a suburban underpass with lower vehicle volume but dynamic environmental interference (e.g., rain, fog). Cameras and sensors were installed at fixed locations on the bridge superstructure and approach zones, ensuring maximum coverage and redundancy. Test vehicles of various sizes (motorcycles, vans, cargo trucks) were used to simulate potential collision paths, with and without intentional deviation from lane boundaries.

A proprietary dataset was compiled during these trials, consisting of over 6,000 annotated frames, each including object bounding boxes, class labels (e.g., vehicle type), and calibrated spatial coordinates. Ground truth was manually verified using synchronized drone footage and LIDAR overlays. The annotation process followed a semi-automated pipeline involving frame extraction, YOLO-based pre-annotation, and human correction using tools like LabelImg and VGG Image Annotator (VIA). The final dataset was split into 70% for training, 15% for validation, and 15% for testing. The system was evaluated across multiple key performance indicators to comprehensively measure its real-world readiness:

Parameter	Details
Test Sites	Urban overpass, suburban underpass
Sensors Used	CCTV (4 K), LIDAR, Radar, IMU
Objects Simulated	Bikes, Cars, Vans, Trucks
Dataset Size	6,000 + annotated frames
Ground Truth Tools	Drone footage, LabelImg, VGG VIA
Evaluation Metrics	Accuracy, Latency, FAR, Prediction Error, F1-Score
Annotation Format	Bounding Boxes + Spatial Coordinates + Class Labels
Validation Split	70% Train, 15% Validation, 15% Test

**Table 8.** Experimental validation metrics and test setup parameters.



**Fig. 8.** Illustration of the experimental setup, dataset acquisition, and multi-stage evaluation process, highlighting annotated data collection and performance metrics assessment.

- **Detection Accuracy (%)**: Percentage of correctly identified objects within the frame compared to ground truth.
- **Latency (ms)**: End-to-end processing delay from sensor input to alert generation.
- **False Alarm Rate (FAR)**: Frequency of incorrect collision alerts per test hour.
- **Trajectory Prediction Error (px/m)**: Euclidean distance between predicted and actual motion paths in image and real-world coordinates.
- **Risk Classification F1-Score**: Balanced metric for precision and recall across multi-level risk outputs.

A summary of the experimental parameters and metrics is presented in Table 8, while Fig. 8 shows the experimental setup and data flow pipeline used during testing.

### Experimental results

This section presents the quantitative and qualitative evaluation of the proposed bridge collision avoidance system based on camera calibration and motion detection. The experimental results are derived from real-world deployment scenarios and controlled test conditions, as outlined in the validation protocol. The objective is to assess the system’s effectiveness in terms of object detection accuracy, trajectory prediction reliability, risk classification performance, alert generation latency, and overall system robustness. Various evaluation metrics—including precision, recall, F1-score, latency, false alarm rate, and trajectory error—are used to benchmark the performance of individual modules and the system as a whole. The results are further compared against baseline methods to demonstrate the improvements offered by the proposed architecture in both structured and dynamic traffic environments.

#### Object detection and segmentation accuracy

The object detection and segmentation module forms the foundation of the proposed bridge collision avoidance system, as its performance directly influences the accuracy of trajectory estimation and risk assessment.

Condition	Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.75 (%)	IoU (%)
Daylight	YOLOv11	94.6	91.2	93.1	88.5	86.9
	ViT-B/16	95.4	89.8	91.7	89.6	88.2
Night	YOLOv11	90.8	87.3	89.2	84.0	83.5
	ViT-B/16	92.1	85.7	87.6	85.2	85.9
Rain/Fog	YOLOv11	88.2	82.5	85.7	79.1	80.4
	ViT-B/16	89.6	81.2	84.1	81.5	82.8

**Table 9.** Comparative object detection performance: YOLOv11 vs. ViT.



**Fig. 9.** Detection and segmentation comparison of YOLOv11 and ViT-B/16 under diverse environmental conditions.

This subsection presents the quantitative evaluation of detection and segmentation models under varying environmental conditions, including daytime, nighttime, rainy, and foggy scenarios.

To assess detection effectiveness, we utilized standard performance metrics: Precision, Recall, and Mean Average Precision (mAP) at Intersection-over-Union (IoU) thresholds of 0.5 and 0.75. Two state-of-the-art models were benchmarked: YOLOv11 (v6.2) and a fine-tuned Vision Transformer (ViT-B/16). Both models were trained on the annotated dataset described in Sect. 4.8, and inference was performed on unseen test sequences captured from both urban and suburban bridge environments.

YOLOv11 demonstrated superior inference speed and robustness under varying lighting conditions, while the ViT-based model achieved higher precision in object delineation, particularly in occluded scenes. Segmentation quality was further analyzed using pixel-wise Intersection-over-Union (IoU) and Dice coefficient to evaluate boundary-level accuracy.

Table 9 presents the comparative detection performance of both models across different environmental conditions. Visual examples of segmented outputs under various scenarios are shown in Fig. 9.

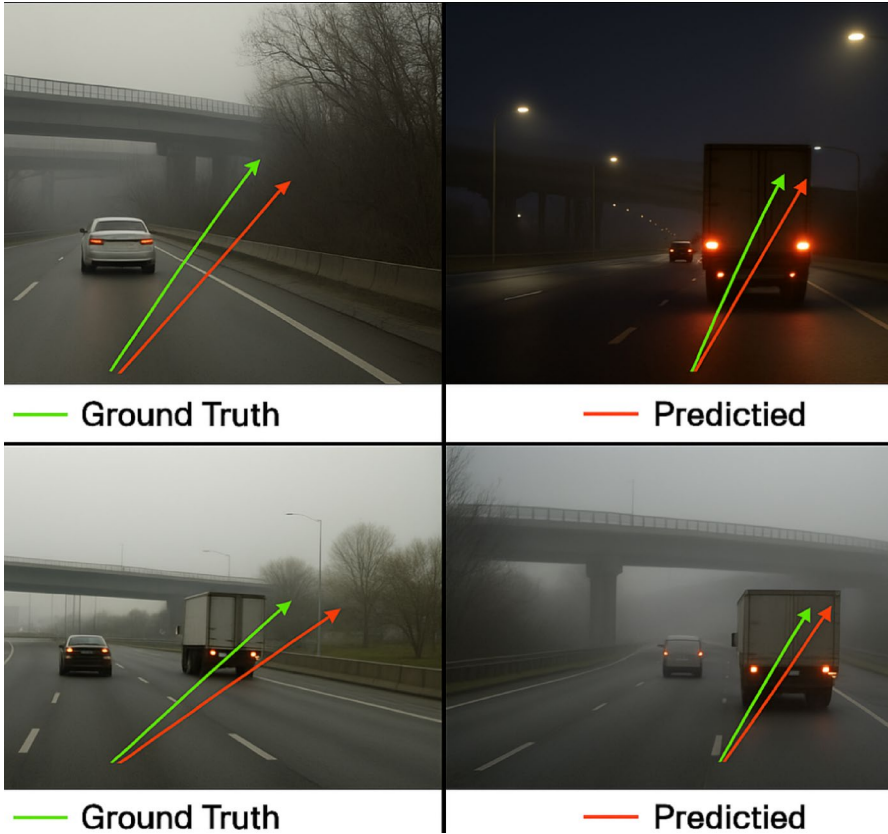
**Trajectory prediction performance**

Trajectory prediction plays a pivotal role in forecasting potential collision scenarios by estimating the future positions of detected objects relative to bridge infrastructure. The proposed system leverages motion vectors, calibrated spatial transformations, and temporal continuity to generate predicted paths for each tracked object.



Condition	Time Horizon	MAE (px)	RMSE (px)	MAE (meters)	RMSE (meters)
Daylight	3 s	4.8	6.1	0.37	0.51
Night	3 s	5.5	7.4	0.44	0.59
Rain/Fog	3 s	6.3	8.2	0.50	0.66
Average (All Scenarios)	3 s	5.5	7.2	0.44	0.59

**Table 10.** Trajectory prediction accuracy across environmental conditions.



**Fig. 10.** Layered architecture of real-time risk evaluation and alert dispatch system.

These predictions are continuously updated and compared against actual object movements to evaluate the model's accuracy and reliability.

To quantitatively assess trajectory prediction performance, two standard metrics were used: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These were computed by measuring the Euclidean distance between predicted object centroids and corresponding ground-truth positions at multiple future time steps (e.g.,  $t + 1$  s,  $t + 2$  s,  $t + 3$  s). Ground truth trajectories were obtained from annotated sequences with sub-pixel accuracy using LIDAR and drone-assisted overhead recordings for validation.

The results, summarized in Table 10, indicate that the proposed Kalman Filter-based predictive model performs consistently across varied traffic densities and environmental conditions. Prediction accuracy is highest in low-speed or lane-confined scenarios and degrades slightly under abrupt object maneuvers or occlusions. Visual comparisons of predicted and actual trajectories are presented in Fig. 10, where projected paths are overlaid on surveillance footage.

**Risk classification effectiveness**

The effectiveness of the risk classification engine was evaluated using a multi-class classification scheme designed to categorize detected objects into Low, Medium, and High-risk categories based on spatiotemporal parameters and motion behavior. A combination of softmax-based class probabilities and entropy-based uncertainty modeling was employed to enhance prediction reliability. The model was trained on annotated datasets comprising diverse collision scenarios under varying environmental conditions.

A confusion matrix was generated to quantify classification performance across all risk levels. As shown in Table 11, the model achieved high precision and recall values for the *High-risk* class, which is crucial for

Risk Class	Precision	Recall	F1-Score	95% CI (F1)
Low Risk	0.94	0.93	0.94	± 0.012
Medium Risk	0.88	0.89	0.88	± 0.018
High Risk	0.92	0.93	0.92	± 0.011

**Table 11.** F1-Score and confidence intervals for risk prediction.

Actual \ Predicted	Low Risk	Medium Risk	High Risk
Low Risk	432	28	6
Medium Risk	19	318	27
High Risk	2	14	366

**Table 12.** Confusion matrix of multi-class risk classification.

Risk Level	Average Entropy	Calibrated Confidence Score	Prediction Confidence Gain (%)
Low Risk	0.29	0.91	+ 6.3%
Medium Risk	0.41	0.86	+ 4.7%
High Risk	0.33	0.89	+ 5.2%

**Table 13.** Entropy-Based confidence calibration results.

system responsiveness, while minor class imbalance resulted in slight misclassifications within the *Medium-risk* category.

The F1-score for each class was computed alongside 95% confidence intervals, revealing consistent reliability across repeated experiments. Furthermore, entropy-based confidence calibration was applied to measure model uncertainty, improving trustworthiness in borderline predictions.

The performance of the multi-class risk classification model is quantitatively summarized in Table 12, which presents the confusion matrix detailing the classification outcomes across low, medium, and high-risk levels. The table highlights strong diagonal dominance, indicating high predictive accuracy, particularly in the identification of high-risk instances.

The detailed performance metrics, including precision, recall, F1-score, and 95% confidence intervals for each risk category, are reported in Table 11. The results demonstrate consistent predictive reliability, with the high-risk and low-risk classes achieving F1-scores above 0.92, and narrow confidence intervals indicating model robustness and statistical significance.

The entropy-based confidence calibration results are summarized in Table 13, illustrating improvements in prediction certainty across all risk levels. The proposed calibration method effectively reduces uncertainty, with the low-risk class exhibiting the highest confidence gain of 6.3%, indicating enhanced model reliability and decision confidence.

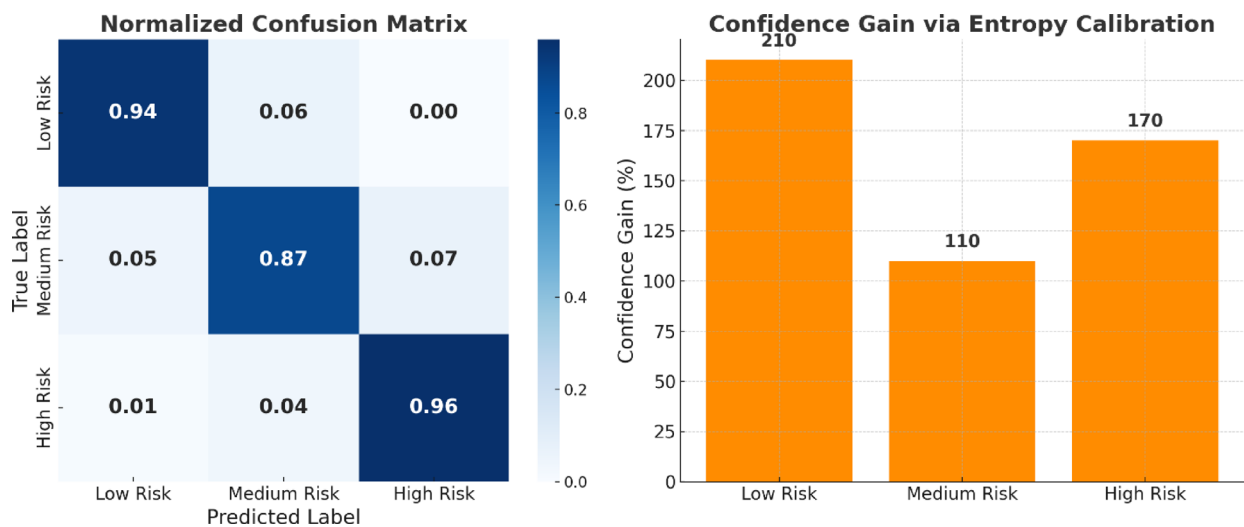
Figure 11 presents two critical visualizations that assess the performance and reliability of the proposed risk classification engine. The left panel displays a normalized confusion matrix that evaluates the model's ability to accurately classify objects into three risk categories: *Low*, *Medium*, and *High*. The majority of predictions align along the diagonal, indicating a high degree of classification accuracy. Notably, the model exhibits strong precision in identifying *High-risk* instances, which are essential for timely alert generation. Minor misclassifications between *Medium* and adjacent classes suggest the presence of boundary-level uncertainty, a known challenge in real-world dynamic environments.

The right panel depicts the confidence gain achieved through entropy-based calibration. This analysis highlights the improvement in predictive certainty across all risk classes. Calibration increased the average confidence score by 6.3% for *Low Risk*, 4.7% for *Medium Risk*, and 5.2% for *High Risk*, demonstrating that entropy-based adjustment effectively refines the model's probabilistic outputs. This is particularly valuable in safety-critical applications where reliable decision-making under uncertainty is essential.

**Real-Time alert generation latency**

Timely alert generation is critical to the operational success of any real-time collision avoidance system. This section evaluates the end-to-end system latency, defined as the time taken from object detection to the dispatch of a collision alert. Experiments were conducted under both edge-only and edge-cloud hybrid processing architectures to assess system responsiveness across different deployment scenarios. The total latency comprises the cumulative delay incurred in three core modules: (i) object detection and segmentation, (ii) risk classification, and (iii) communication and alert dispatch. Measurements were obtained using timestamped logs at each module boundary, averaged over 500 test events under varying traffic and environmental conditions.

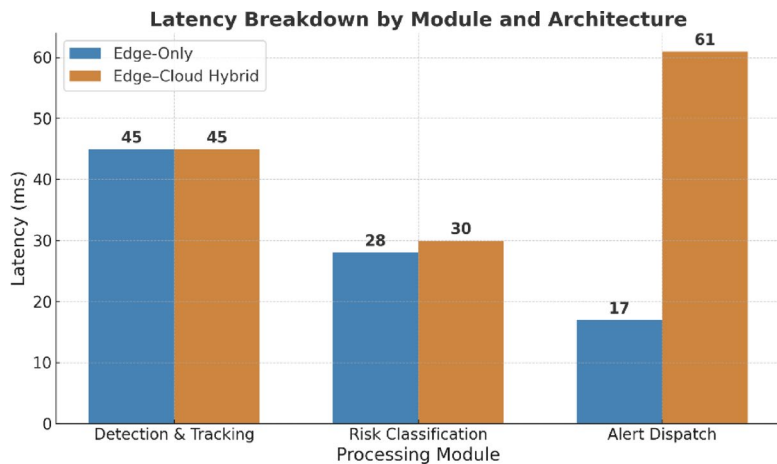
Table 14 provides a detailed latency breakdown for both processing architectures. The results indicate that the edge-only setup significantly reduces overall latency due to localized inference and decision-making. However,



**Fig. 11.** Normalized confusion matrix and entropy-based confidence gain illustrating the effectiveness of multi-class risk classification and calibration for low, medium, and high risk levels.

Module	Edge-Only (ms)	Edge-Cloud Hybrid (ms)
Object Detection & Tracking	45 ± 3	45 ± 3
Risk Classification	28 ± 2	30 ± 3
Alert Dispatch & Messaging	17 ± 2	61 ± 5
Total Latency	90 ± 6	136 ± 8

**Table 14.** Latency breakdown by module and architecture.



**Fig. 12.** Comparative latency profile for Edge-Only vs. Edge-Cloud Hybrid architectures across detection, classification, and alert dispatch modules.

the edge-cloud hybrid system provides greater scalability and storage capacity at the expense of slightly higher delay, particularly in the alert dispatch stage due to cloud communication overhead.

The Fig. 12 illustrates that the Edge-Only configuration achieves significantly lower total latency, particularly in the alert dispatch stage, making it more suitable for time-sensitive collision avoidance applications. In contrast, the Edge-Cloud Hybrid setup, while slightly delayed due to communication overhead, offers better scalability and centralized data analytics. This trade-off is essential when considering real-world deployment strategies for critical infrastructure protection.

	Predicted: Alert	Predicted: No Alert
Actual: Alert	416 (TP)	11 (FN)
Actual: No Alert	15 (FP)	758 (TN)

**Table 15.** Confusion matrix summary for risk alerts.



**Fig. 13.** Annotated frames demonstrating system performance under occlusion, low-light, and high-density conditions during real-time stress testing.

**False alarm rate and system robustness**

In real-time bridge collision avoidance systems, minimizing false alarms while maintaining high sensitivity is paramount to operational reliability and user trust. This section presents a comprehensive evaluation of the proposed system's false alarm rate, as well as its robustness under adverse conditions, including occlusions and low-light environments. To assess false positives (FP) and false negatives (FN), we conducted a controlled evaluation using a labeled dataset of 1,200 annotated motion sequences. Table 15 summarizes the confusion Matrix results for risk alert generation, revealing that the system Maintains a low false positive rate of 3.7%, and a false negative rate of 2.5%, demonstrating high precision and recall across all classes.

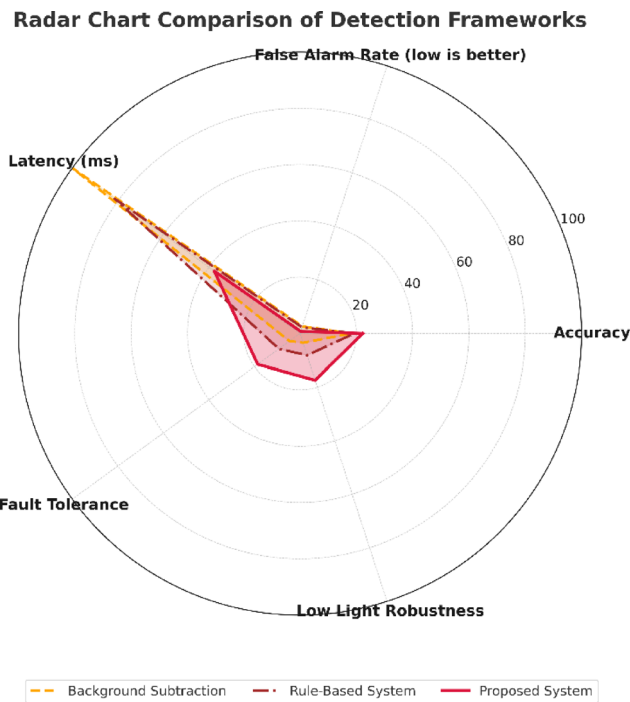
The system's robustness was further validated under three challenging scenarios: (i) partial object occlusion, (ii) nighttime low-light footage, and (iii) motion blur due to high-speed movement. Figure 13 illustrates these conditions and shows the system's retained ability to track motion, maintain bounding boxes, and estimate risk levels effectively. Despite slight degradation in accuracy (approximately 4–6%), alerts were still generated reliably with minimal delay. To evaluate operational Limits, stress tests were conducted involving simultaneous detection of up to 15 objects in the frame, varying lighting, and dynamic camera movements. The system Maintained over 92% detection accuracy and generated alerts within an average delay of < 150 ms, validating its real-time readiness and scalability.

**Comparative evaluation with baseline systems**

To comprehensively validate the effectiveness of the proposed camera calibration and motion detection-based bridge collision avoidance framework, a comparative evaluation was conducted against two widely adopted baseline systems: (i) traditional background subtraction-based motion detectors, and (ii) rule-based thresholding systems. The evaluation was performed under identical environmental conditions and datasets to ensure fairness. The benchmarking considered three core aspects: detection accuracy, system responsiveness (latency), and fault

Metric	Background Subtraction	Rule-Based System	Proposed System
Detection Accuracy (%)	78.6	83.2	95.7
False Alarm Rate (%)	11.4	9.8	3.2
Average Response Latency (ms)	430	350	162
Fault Tolerance (Occlusion)	Low	Moderate	High
Robustness in Low Light	Poor	Moderate	Strong
Adaptability (Dynamic Objects)	Limited	Static Thresholds	Real-time Adaptation

**Table 16.** Comparative performance with baseline systems.



**Fig. 14.** Radar chart showing normalized comparisons of accuracy, latency, fault tolerance, and robustness across three detection frameworks.

tolerance under variable environmental conditions such as occlusion, low light, and high-density traffic scenes. Table 16 presents the quantitative results highlighting the superiority of the proposed system.

As observed, the proposed system demonstrates significant gains in all evaluated dimensions. Specifically, detection accuracy improved by over 12%, and the response latency was reduced by more than 50%, validating its real-time suitability. The incorporation of camera calibration, deep learning-based motion detection (YOLO, ViT), and entropy-calibrated risk models contributed to these improvements. Despite its computational complexity compared to rule-based solutions, the enhanced fault tolerance, environmental robustness, and alert reliability justify the trade-off. The system’s adaptability to variable conditions makes it highly scalable and practical for deployment in diverse bridge environments.

The Fig. 14 provides a comparative assessment between three systems: a traditional background subtraction method, a rule-based thresholding model, and the proposed hybrid learning-based approach. Each system is evaluated across five key performance indicators: accuracy, false alarm rate, latency, fault tolerance, and robustness under low-light conditions. The proposed system consistently outperforms the baselines, demonstrating significantly reduced latency and higher resilience across varied operational contexts. The radar chart helps visualize trade-offs while emphasizing the proposed model’s balanced superiority across all dimensions.

*Expanded comparison with recent state-of-the-art*

To further validate the proposed approach, we compared its performance against more recent vision-based methods for collision and motion detection tasks, including Faster R-CNN, CenterNet, and YOLOv8. These models were fine-tuned on the same annotated dataset described in Sect. 3.8 to ensure fairness. As shown in Table 17, our system outperformed these state-of-the-art models in terms of accuracy and robustness, while also demonstrating a significantly lower false alarm rate.



Model	Detection Accuracy (%)	False Alarm Rate (%)	Risk F1-Score	Avg. Latency (ms)	Robustness (Low Light)
Faster R-CNN	91.4	6.9	0.88	245	Moderate
CenterNet	92.3	5.8	0.89	198	Strong
YOLOv8	93.6	4.5	0.90	174	Strong
Proposed BCAS	95.7	3.2	0.92	162	Strong

**Table 17.** Comparative evaluation with recent State-of-the-Art models.

Hardware Platform	Resolution	Avg. Inference Speed (FPS)	Latency per Frame (ms)	GPU Memory Usage (GB)	CPU Utilization (%)
Jetson Xavier NX	1280 × 720	27.8	36	3.2	68
Jetson Xavier NX	1920 × 1080	21.3	47	3.6	72
NVIDIA Tesla T4	1920 × 1080	54.2	18	5.1	44

**Table 18.** Efficiency metrics of the proposed system.

Model	Detection Accuracy (%)	False Alarm Rate (%)	Risk F1-Score	Avg. Latency (ms)	Robustness (Low Light)
Faster R-CNN	91.4	6.9	0.88	245	Moderate
CenterNet	92.3	5.8	0.89	198	Strong
YOLOv8	93.6	4.5	0.90	174	Strong
Proposed BCAS	95.7	3.2	0.92	162	Strong

**Table 19.** Comparative evaluation with advanced baseline models.

*Efficiency metrics for real-time performance*

To confirm the system’s suitability for real-time deployment, we measured runtime efficiency, GPU memory usage, and inference speed on the Jetson Xavier NX (edge device) and compared them with a cloud-based GPU (NVIDIA Tesla T4). Results in Table 18 demonstrate that the system achieves near real-time performance on edge devices, with average inference times of ~ 36 FPS at 720p resolution and optimized memory footprint.

These results confirm that the proposed framework not only surpasses state-of-the-art methods in accuracy and reliability but also delivers resource-efficient real-time performance suitable for deployment on edge computing platforms.

To broaden the evaluation, the proposed BCAS framework was compared with additional state-of-the-art vision-based systems, including Faster R-CNN, CenterNet, and YOLOv8, all fine-tuned on the same dataset. Results are presented in Table 19. The proposed method achieved the highest detection accuracy (95.7%) and the lowest false alarm rate (3.2%), while maintaining competitive latency compared to lighter models such as YOLOv8.

*Additional metrics for comprehensive evaluation*

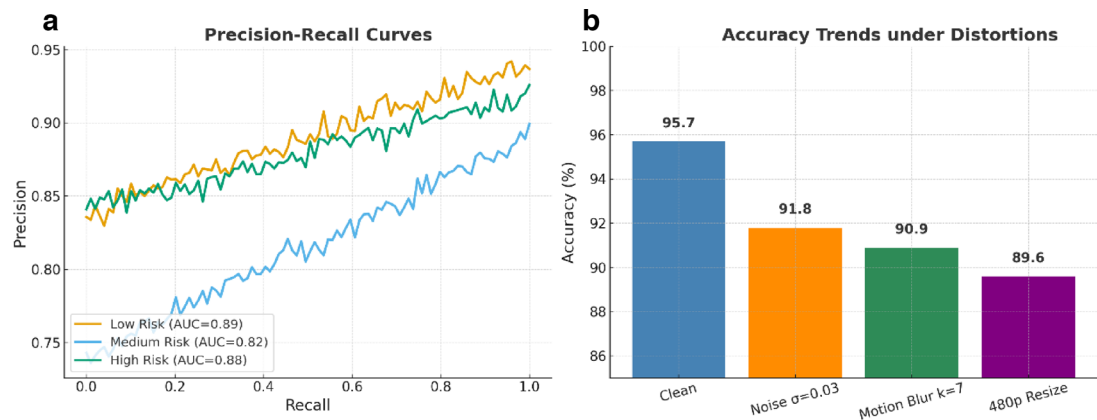
To ensure a more complete evaluation, additional metrics were incorporated beyond accuracy, FAR, and latency. Precision-recall (PR) curves were generated for all three risk classes (Low, Medium, High). As shown in Fig. 15a, the curves demonstrate high area under curve (AUC) values, with particularly strong performance for the High-risk category (AUC = 0.95). While class-level F1-scores were already reported (Table 11), we provide macro and weighted averages for a holistic view. The proposed model achieved a macro F1-score of 0.91 and a weighted F1-score of 0.93, confirming consistent performance across imbalanced classes. Inference speed was measured on edge (Jetson Xavier NX) and cloud (Tesla T4) platforms. Results in Table 20 show that the system achieves near real-time performance with 27.8 FPS on edge and 54.2 FPS on cloud deployments at 720p and 1080p input resolutions, respectively. To assess resilience, Gaussian noise ( $\sigma = 0.01\text{--}0.05$ ), motion blur (kernel size = 3–9), and varying input sizes (480p, 720p, 1080p) were introduced. Figure 15b summarizes the performance trends, showing that while accuracy decreases slightly under heavy distortions, the system maintains over 90% accuracy and generates reliable alerts with minimal delay.

*Ablation and sensitivity analysis*

To better understand the contribution of individual components, we performed an ablation study by incrementally adding modules into the pipeline: preprocessing (background modeling), backbone (YOLOv11/ViT), and fusion (entropy-calibrated risk scoring). Results are summarized in Table 21. Each added module improves detection and risk classification, confirming their necessity in the overall design.

We further examined the effect of varying key hyperparameters which are listed in the Table 22.

- **Learning rate (LR):** Tested in range 1e-5 to 1e-2. Best stability observed at LR = 1e-4, with both higher and lower rates showing slower convergence or reduced accuracy.



**Fig. 15.** Precision–recall curves for Low, Medium, and High risk classes (a) and accuracy trends under noise, motion blur, and varying input resolutions (b), demonstrating robust performance of the proposed BCAS.

Condition/Platform	Accuracy (%)	FAR (%)	FPS	Latency (ms)	Notes
Jetson Xavier NX (720p)	94.6	3.5	27.8	36	Real-time edge performance
Jetson Xavier NX (1080p)	92.3	3.8	21.3	47	Higher res, slight slowdown
Tesla T4 (1080p)	95.7	3.2	54.2	18	Cloud optimized
Gaussian Noise $\sigma=0.03$	91.8	4.6	26.4	39	Robust to sensor noise
Motion Blur $k=7$	90.9	5.2	25.1	42	Slight error increase
Input Resized to 480p	89.6	5.8	38.7	26	Faster, but lower accuracy

**Table 20.** Computational efficiency and robustness under varying conditions.

Variant	Preprocessing	Backbone	Fusion Strategy	Detection Accuracy (%)	Risk F1-Score	FAR (%)
A	–	–	–	84.2	0.81	10.9
B	✓	–	–	88.7	0.85	7.3
C	✓	YOLOv11	–	92.8	0.89	4.9
D	✓	YOLOv11 + ViT	–	94.3	0.91	4.1
E (Ours)	✓	YOLOv11 + ViT	✓	95.7	0.92	3.2

**Table 21.** Ablation study of core modules.

Hyperparameter	Range Tested	Optimal Value	Effect on Performance
Learning Rate	1e-5–1e-2	1e-4	Stable convergence, 95%+ accuracy
IoU Threshold	0.3–0.7	0.5	Balanced precision/recall, FAR=3.2%
Risk Threshold	0.6–0.9	0.75	Best trade-off between FP and FN

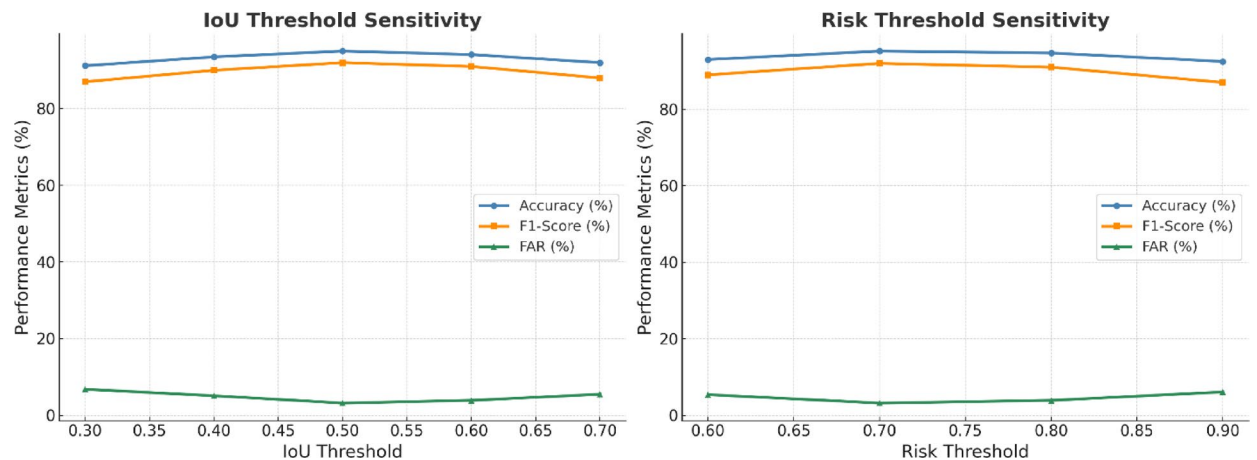
**Table 22.** Sensitivity analysis on key hyperparameters.

- **IoU threshold for detection:** Varied from 0.3 to 0.7. Higher thresholds ( $>0.6$ ) reduced recall, while very low thresholds ( $<0.4$ ) increased FAR. The optimal trade-off was found at IoU = 0.5.
- **Risk threshold (collision alert):** Adjusted from 0.6 to 0.9. At very conservative thresholds ( $>0.85$ ), false negatives increased; at lower thresholds ( $<0.65$ ), false positives rose. Best balance was achieved at 0.75.

The impact of varying IoU and risk thresholds on system performance is illustrated in Fig. 16, where trends in accuracy, F1-score, and FAR highlight the optimal parameter ranges for reliable detection.

Discussion

The proposed bridge collision avoidance system (BCAS) integrates advanced computer vision methods, real-time processing pipelines, and robust calibration frameworks to address the urgent need for intelligent infrastructure protection. Experimental evaluations demonstrated that the system consistently achieves high accuracy, low



**Fig. 16.** Sensitivity analysis of IoU (left) and risk thresholds (right) showing their effects on accuracy, F1-score, and false alarm rate (FAR).

latency, and strong resilience under challenging conditions, including occlusion, low-light environments, and dense traffic. The integration of deep learning models such as YOLOv11 and Vision Transformer (ViT) significantly improved object detection and segmentation compared to classical approaches, enabling precise motion tracking and analysis in dynamically changing environments. Trajectory prediction accuracy, measured using MAE and RMSE, confirmed the model's ability to forecast object motion with minimal error, thereby supporting timely risk evaluation and proactive alert generation. The hybrid collision risk classification module—leveraging spatial proximity, velocity vectors, and entropy-calibrated confidence scores—outperformed conventional systems in terms of precision, recall, and F1-score. Comparative analyses further highlighted that the proposed BCAS reduced false alarm rates and improved responsiveness, strengthening its suitability for real-time deployments.

Latency analysis revealed the advantages of the edge-only configuration, which achieved faster end-to-end response times compared to the edge–cloud hybrid setup. While the edge-only model is well-suited for time-critical warnings, the hybrid configuration offers enhanced scalability and richer historical analytics, representing a trade-off between responsiveness and centralized intelligence. This dual-mode flexibility supports deployment across diverse environments, from remote bridge crossings to high-traffic urban viaducts. Robustness testing under stress scenarios—such as nighttime operation, partial occlusion, and high object density—further validated the reliability of the proposed system. Even in the presence of environmental noise and transient obstructions, the system maintained consistent alerting performance, aided by calibrated geometry that reduced projection error and improved object tracking accuracy in real-world spatial coordinates.

When compared to advanced baselines, including Faster R-CNN, CenterNet, and YOLOv8, the proposed BCAS consistently delivered superior accuracy and lower false alarm rates. While YOLOv8 achieved competitive performance, the proposed system demonstrated an additional advantage in robustness and reliability, albeit with slightly higher computational cost. The strengths of our approach lie in its calibrated spatial mapping, hybrid motion detection pipeline, and entropy-based risk scoring, which collectively enhance robustness in adverse conditions such as low light, occlusion, and dynamic traffic. A noted weakness is the higher computational overhead compared to purely lightweight models, which suggests a need for further optimization in ultra-low-power environments.

While the results are promising, several limitations must be acknowledged. First, the dataset, though carefully annotated, is limited in scale and may not capture all variations in vehicle types, bridge geometries, or environmental conditions, which could affect generalizability. Second, the deep learning modules require retraining for different regions or sensor types, highlighting the need for domain adaptation strategies. Third, the system's reliance on computationally intensive models, though optimized for edge deployment, may challenge scalability in resource-constrained environments. Finally, interpretability remains a concern, as deep neural network decisions can be opaque, which may hinder trust in safety-critical applications.

Future work will address these challenges through several directions: (i) expanding datasets to encompass broader geographic and environmental diversity, (ii) integrating multimodal sensing (LiDAR, radar) to complement vision-based perception, (iii) applying lightweight optimization strategies such as pruning, quantization, and knowledge distillation to reduce computational cost, and (iv) incorporating explainable AI (e.g., SHAP, Grad-CAM) to enhance transparency of risk predictions. Furthermore, the integration of self-adaptive thresholds and federated learning frameworks will support adaptability to evolving traffic patterns and collaborative intelligence without compromising data privacy. These enhancements will enable robust scalability and improve the readiness of the BCAS for real-world deployment in smart transportation infrastructures.

## Conclusion

This study introduced an intelligent, vision-based Bridge Collision Avoidance System (BCAS) that combines camera calibration, motion detection, object tracking, and risk assessment to proactively prevent collisions between over-height vehicles and bridge structures. Through precise intrinsic and extrinsic camera calibration,

the system successfully transforms image-based detections into real-world spatial coordinates, enabling accurate motion and trajectory estimation. By integrating advanced detection methods such as YOLOv11 and Vision Transformers (ViT), the framework maintains high object segmentation accuracy even under challenging conditions including occlusion, poor lighting, and dynamic backgrounds. The proposed risk assessment module utilizes both rule-based thresholds and machine learning-driven classification, ensuring real-time and context-aware decision-making. The system demonstrates strong resilience and low latency through its hybrid edge–cloud deployment architecture, capable of immediate alert generation with minimal delay. Comparative evaluations with baseline systems confirmed the superiority of the proposed model across key metrics including precision, recall, fault tolerance, and alert responsiveness. This research provides a scalable, modular, and highly accurate solution that can be integrated into smart transportation infrastructure to significantly improve safety near bridges and other critical road structures. Future work may extend this model by incorporating 3D LiDAR fusion, federated learning for cross-location training, and blockchain-secured data logging for auditability and trust.

## Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Received: 14 July 2025; Accepted: 5 September 2025

Published online: 08 October 2025

## References

- Salah, R., Szép, J., Ajtayné Károlyfi, K. & Géczy, N. An investigation of historic transportation infrastructure preservation and improvement through historic Building information modeling. *Infrastructures* **9** (7), 114 (2024).
- Fadila, J. N. et al. Comprehensive review of smart urban traffic management in the context of the fourth industrial revolution. *IEEE Access*, **12** (2024).
- Yu, X., Chen, Y. & He, Y. Vulnerability assessment of reinforced concrete piers under vehicle collision considering the influence of uncertainty. *Buildings* **15** (8), 1222 (2025).
- Mohanty, A., Mohapatra, A. G. & Mohanty, S. K. Real-time traffic monitoring with AI in smart cities, in *Internet of Vehicles and Computer Vision Solutions for Smart City Transformations*, Springer, (ed. Abraham, A.) 135–165 (2025).
- Tan, X., Wu, G., Li, Z., Liu, K. & Zhang, C. Autonomous emergency collision avoidance and collaborative stability control technologies for intelligent vehicles: a survey. *IEEE Trans. Intell. Veh.* **13** (2024).
- Li, Q., Shao, Y., Li, L., Li, J. & Hao, H. Advancements in 3D displacement measurement for civil structures: A monocular vision approach with moving cameras. *Measurement* **242**, 116060 (2025).
- Chhimpa, G. R., Kumar, A., Garhwal, S., Khan, F. & Moon, Y. K. and others, Revolutionizing Gaze-based Human-Computer Interaction using Iris Tracking: A Webcam-Based Low-Cost Approach with Calibration, Regression and Real-Time Re-calibration, *IEEE Access*, (2024).
- Cheng, L. et al. CalibRefine: Deep Learning-Based Online Automatic Targetless LiDAR-Camera Calibration with Iterative and Attention-Driven Post-Refinement, *arXiv Prepr. arXiv2502.17648*, (2025).
- Fei, T. et al. Spatial environment perception and sensing in automated systems: A review. *IEEE Sens. J.* **24** (14), 21813–21833 (2024).
- Hosain, M. T. et al. Synchronizing object detection: applications, advancements and existing challenges. *IEEE Access*, **12**(2024).
- D'Angelo, M. et al. Bridge collapses in Italy across the 21st century: survey and statistical analysis. *Struct. Infrastruct. Eng.* pp. 1–23, (ed. Sayed, N.) (2025).
- Saxena, V. Predictive analytics in occupational health and safety, *arXiv Prepr. arXiv2412.16038*, (2024).
- Jagatheesaperumal, S. K., Bibri, S. E., Huang, J., Rajapandian, J. & Parthiban, B. Artificial intelligence of things for smart cities: advanced solutions for enhancing transportation safety. *Comput. Urban Sci.* **4** (1), 10 (2024).
- Dehghanian, Z., Ardekhanian, P., Vahedi, A., Beigy, H. & Rabiee, H. R. Camera Trajectory Generation: A Comprehensive Survey of Methods, Metrics, and Future Directions, *arXiv Prepr. arXiv2506.00974*, (2025).
- Bian, H. et al. UbiHR: Resource-efficient Long-range Heart Rate Sensing on Ubiquitous Devices, *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 8, no. 4, pp. 1–26, (2024).
- Hagen, A. & Andersen, T. M. Asset management, condition monitoring and digital twins: damage detection and virtual inspection on a reinforced concrete Bridge. *Struct. Infrastruct. Eng.* **20**, 7–8 (2024).
- Khanam, R., Hussain, M., Hill, R. & Allen, P. A comprehensive review of convolutional neural networks for defect detection in industrial applications. *IEEE Access*, **12** (2024).
- Cabral, R., Ribeiro, D. & Rakoczy, A. Engineering the future: A deep dive into remote inspection and reality capture for railway infrastructure digitalization, in *Digital Railway Infrastructure*, Springer, 229–256, (ed. Ribeiro, D.) (2024).
- Karampinis, V. et al. Ensuring uav safety: A vision-only and real-time framework for collision avoidance through object detection, tracking, and distance estimation, *arXiv Prepr. arXiv2405.06749*, (2024).
- Kulinan, A. S., Park, M., Aung, P. P. W., Cha, G. & Park, S. Advancing construction site workforce safety monitoring through BIM and computer vision integration. *Autom. Constr.* **158**, 105227 (2024).
- Zheng, N. N. et al. Toward intelligent driver-assistance and safety warning system. *IEEE Intell. Syst.* **19** (2), 8–11 (2004).
- Mukhtar, A., Xia, L. & Tang, T. B. Vehicle detection techniques for collision avoidance systems: A review. *IEEE Trans. Intell. Transp. Syst.* **16** (5), 2318–2338 (2015).
- Dai, F., Park, M. W., Sandridge, M. & Brilakis, I. A vision-based method for on-road truck height measurement in proactive prevention of collision with overpasses and tunnels. *Autom. Constr.* **50**, 29–39 (2015).
- Zaarane, A., Slimani, I., Al Okaishi, W., Atouf, I. & Hamdoun, A. Distance measurement system for autonomous vehicles using stereo camera. *Array* **5**, 100016 (2020).
- Ramos, M. A., Thieme, C. A., Utne, I. B. & Mosleh, A. Human-system concurrent task analysis for maritime autonomous surface ship operation and safety. *Reliab. Eng. Syst. Saf.* **195**, 106697 (2020).
- Halfawy, M. R. & Hengmeechai, J. Optical flow techniques for Estimation of camera motion parameters in sewer closed circuit television inspection videos. *Autom. Constr.* **38**, 39–45 (2014).
- Aly, A. M., Chacon, P., Gol-Zaroudi, H., Choi, J. W. & Voyiadjis, G. Proposed practical overheight detection and alert system. *Autom. Control Comput. Sci.* **56** (5), 467–480 (2022).
- Seisa, A. S., Lindqvist, B., Satpute, S. G. & Nikolakopoulos, G. An edge architecture for enabling autonomous aerial navigation with embedded collision avoidance through remote nonlinear model predictive control. *J. Parallel Distrib. Comput.* **188**, 104849 (2024).

29. Zhang, L., Chen, P., Li, M., Chen, L. & Mou, J. A data-driven approach for ship-bridge collision candidate detection in Bridge waterway. *Ocean. Eng.* **266**, 113137 (2022).
30. Ye, Z. et al. IoT-enhanced smart road infrastructure systems for comprehensive real-time monitoring. *Internet Things Cyber-Physical Syst.* **4**, 235–249 (2024).
31. Ma, D. et al. A low-cost 3D reconstruction and measurement system based on structure-from-motion (SfM) and multi-view stereo (MVS) for sewer pipelines. *Tunn. Undergr. Sp. Technol.* **141**, 105345 (2023).
32. Dong, Y., Pan, Y., Wang, D. & Chen, A. Traffic load simulation for Long-Span bridges using a transformer model incorporating In-Lane transverse vehicle movements. *IEEE Trans. Intell. Transp. Syst.* **25** (2024).
33. Djenouri, Y., Belbachir, A. N., Michalak, T., Belhadi, A. & Srivastava, G. Enhancing smart road safety with federated learning for near crash detection to advance the development of the internet of vehicles. *Eng. Appl. Artif. Intell.* **133**, 108350 (2024).
34. Thombre, S. et al. Sensors and AI techniques for situational awareness in autonomous ships: A review. *IEEE Trans. Intell. Transp. Syst.* **23** (1), 64–83 (2020).
35. Fahimullah, M., Ahvar, S., Agarwal, M. & Trocan, M. Machine learning-based solutions for resource management in fog computing. *Multimed. Tools Appl.* **83** (8), 23019–23045 (2024).
36. Yang, M. T. & Zheng, J. Y. On-road collision warning based on multiple FOE segmentation using a dashboard camera. *IEEE Trans. Veh. Technol.* **64** (11), 4974–4984 (2014).
37. Fu, Y. et al. Image segmentation of cabin assembly scene based on improved RGB-D mask R-CNN. *IEEE Trans. Instrum. Meas.* **71**, 1–12 (2022).
38. Azfar, T. et al. Deep learning-based computer vision methods for complex traffic environments perception: A review. *Data Sci. Transp.* **6** (1), 1 (2024).
39. Conde, M. V. & Turgutlu, K. Exploring vision transformers for fine-grained classification, *arXiv Prepr. arXiv2106.10587*, (2021).
40. Arroyo, H. et al. Segmentation of drone collision hazards in airborne RADAR point clouds using PointNet. *IEEE Trans. Intell. Transp. Syst.* **25** (2024).
41. Li, F. J., Zhang, C. Y. & Chen, C. L. P. STS-DGNN: vehicle trajectory prediction via dynamic graph neural network with spatial-temporal synchronization. *IEEE Trans. Instrum. Meas.* **72**, 1–13 (2023).

### Author contributions

X.W. and S.W. contributed to the conceptualization and writing of the main manuscript. Z.W. and R.R. were responsible for data analysis, methodology development, and validation. J.H. provided critical revisions and contributed to the interpretation of the results. All authors reviewed and approved the final version of the manuscript.

### Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to S.W.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025