Journal on
Artificial Intelligence

Tech Science Press

https://doi.org/10.32604/jai.2026.075753

ARTICLE

Camera-LiDAR Fusion for Enhanced Object Detection

Jianping Wu[1], **Nian Li**[2,*], **Libin Dong**[3] **and Ping Zhang**[4]

[1]School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, China
[2]School of Physics and Electronic Science, Changsha University of Science and Technology, Changsha, China
[3]Zhejiang Datang Wushashan Power Generation Co., Ltd., Ningbo, China
[4]Fuel Business Division, Hunan Datang Xianyi Technology Co., Ltd., Changsha, China

*Corresponding Author: Nian Li. Email: echo20240906@gmail.com

Received: 07 November 2025; Accepted: 19 January 2026; Published: 12 May 2026

ABSTRACT: This paper presents a static fusion framework that enhances object detection by integrating camera and LiDAR-based detection results. The proposed method focuses on associating 2D candidate bounding boxes from a camera detector with 3D candidate boxes from a LiDAR detector using an Intersection over Union (IoU)-based matching approach. To enhance the quality of 2D detection, we refine the baseline Cascade R-CNN detector by incorporating a dual self-attention mechanism into both the backbone and the region proposal network (RPN), resulting in the DA-Cascade R-CNN. This enhancement strengthens the network's ability to detect small or distant objects by improving feature sensitivity and localization accuracy. Once 2D and 3D candidate boxes are obtained, they are associated through IoU-aware matching and subsequently refined using non-maximum suppression (NMS) to remove redundant or conflicting hypotheses across modalities, effectively preserving positive detection results to improve accuracy. Experimental results on the KITTI dataset demonstrate that the proposed static fusion method yields improved detection average precision for three different levels of difficulty compared to single-sensor baselines.

KEYWORDS: Camera object detection; LiDAR object detection; fused object detection; attention mechanism

1 Introduction

With the rise of artificial intelligence and deep learning, object detection has emerged as a fundamental research area in computer vision. In particular, 3D object detection [1,2] has attracted increasing attention due to its ability to detect and localize objects in three-dimensional space. Unlike traditional 2D object detection that identifies objects in the image plane, 3D object detection further estimates the spatial position, size, and orientation of each object, providing richer scene understanding. Typical data sources for 3D object detection include LiDAR point clouds, stereo camera images, and multi-view imagery, all of which provide essential depth cues to improve accuracy in detection and localization. This technology plays a crucial role in a wide range of applications, including autonomous driving, robotics, augmented reality (AR), and virtual reality (VR), where precise and robust environmental perception is required.

In autonomous driving, single-sensor object detection has inherent limits. Vision-only systems lose accuracy at night or under glare because imaging degrades. LiDAR-only systems are not affected by illumination, but rain and snow cause scattering that markedly lowers point-cloud quality. To meet stringent reliability and accuracy requirements, academia and industry use multi-sensor fusion. Camera texture and semantics complement LiDAR geometric depth, improving perception accuracy and robustness.

Sensor-fusion schemes are commonly grouped into early fusion, mid-level feature fusion, and late fusion. Early fusion methods attach image semantics to LiDAR features to learn joint representations [3], while mid-level approaches seed and refine 3D proposals with cross-modal cues [4]. In contrast, late fusion operates on the output of independent detectors. Common techniques for merging detection boxes from different sensors include Weighted Box Fusion (WBF) [5] and Non-Maximum Suppression (NMS) [6]. While standard NMS is widely used in leading detectors such as Faster R-CNN [7], YOLO [8], and Single Shot MultiBox Detector, SSD [9], it has well-documented limitations, such as the potential suppression of valid detections for closely located objects and high sensitivity to confidence score ranking and the Intersection over Union (IoU) threshold. More recent methods designed to strengthen cross-modal interaction by using transformer-style association [10] or by lifting images to Bird's-Eye View (BEV) [11].

Multi-sensor fusion aims to overcome the limits of single sensors and to enhance overall detection performance and robustness. Prior works have demonstrated that fusing camera and LiDAR features can improve detection performance, particularly for long-range objects and complex scenes [12]. Motivated by these observations, this paper proposes a decision-level Camera-Lidar fusion detection algorithm that combines outputs from LiDAR and camera detectors. Instead of directly fusing heterogeneous features, the proposed method emphasizes improving the detection capability of each modality before fusion. Specifically, to overcome the limitations of Cascade R-CNN [13] in detecting small or distant objects, we introduce DA-Cascade R-CNN, a novel architecture that integrates dual self-attention modules into both the backbone and the region proposal network (RPN). This design strengthens the model's ability to capture long-range contextual dependencies, thereby improving the quality of 2D candidate detections. Following independent detection, the 2D outputs from DA-Cascade R-CNN are associated with the 3D bounding boxes generated by a LiDAR-based detector using the Intersection over Union (IoU) metric. A robust fusion strategy is then applied using an IoU-aware Non-Maximum Suppression (NMS) scheme. Unlike standard NMS and Soft-NMS [14], the proposed IoU-aware NMS explicitly incorporates cross-modal spatial overlap, effectively suppressing redundant detections while preserving high-confidence true positives. Moreover, this strategy maintains low computational complexity, making it particularly suitable for scenarios with sparse object distributions, such as the KITTI dataset. Experimental results demonstrate that the proposed fusion detection method significantly improves the detection average precision compared to the single-sensor approach in three different levels of object detection tasks.

2 Related Work

This study aims to develop a fused robust camera–LiDAR object detection framework. To achieve this, we first analyze the framework's core components and their evaluation criteria. This section systematically reviews three areas: advanced 2D image object detectors and the rationale for selecting Cascade R-CNN as the base for our improvements; mainstream 3D point-cloud detection methods and the case for using SECOND [15] as the LiDAR perception module; and the KITTI benchmark dataset for model training and performance evaluation. This review lays a solid theoretical and practical foundation for the fusion method and optimization strategies developed in the following sections.

2.1 Camera Object Detection: Cascade R-CNN

Camera-based object detection [16] is a core computer-vision technique that identifies and localizes objects in images. Among various methods, Cascade R-CNN [13] is an efficient framework built upon the classical two-stage detector Faster R-CNN [7]. While single-stage detectors such as YOLOv5 [17] offer high inference speed, they often struggle with high-precision localization in cluttered autonomous driving scenes. Cascade R-CNN is selected as our baseline because its multi-stage refinement architecture naturally

addresses the quality mismatch problem between training IoU and proposal quality, providing a more stable foundation for the subsequent dual self-attention enhancements. Its key strength is a cascade of detection heads. Multiple heads are connected in series, and training raises the IoU threshold step by step—such as 0.5, 0.6, and 0.7—to redefine positive and negative samples. The regression output of each stage supplies higher-quality proposals to the next stage. This progressive refinement mitigates the quality mismatch between proposal boxes and the IoU used to train the heads, and it markedly improves bounding-box localization. Therefore, this paper adopts Cascade R-CNN as the base camera detector.

2.2 LiDAR Point-Cloud Object Detection: SECOND

LiDAR [18] identifies and localizes objects by generating 3D point cloud data. Among point-cloud detectors, voxel-based methods are widely used because they efficiently process 3D data. This paper adopts the SECOND [15] algorithm—short for Sparsely Embedded Convolutional Detection—as a representative voxel-based approach. It can be viewed as an optimized version of VoxelNet [19]. By introducing sparse convolution to operate on voxelized point clouds, we greatly improved the efficiency, making it suitable for real-time applications. The SECOND pipeline comprises three modules: point-cloud voxelization with feature extraction; sparse-convolution middle-feature extraction; and a Region Proposal Network, RPN.

2.3 Dataset: KITTI

This study utilizes the KITTI [20] dataset, a leading benchmark for autonomous driving, for training and validation. It was created by the Karlsruhe Institute of Technology and the Toyota Technological Institute at Chicago, and contains rich real-world driving scenes. KITTI provides synchronized LiDAR point clouds and camera images, along with detailed sensor calibration, covering urban roads and other scenarios. For standardized evaluation, KITTI defines three difficulty levels—easy, moderate, and hard—based on object occlusion and truncation.

3 Overall Framework

This paper proposes a fused detection algorithm based on camera and LiDAR. The framework is shown in Fig. 1. Given image data and LiDAR data, the algorithm first runs an optimized Cascade R-CNN on images and SECOND on point clouds to obtain 2D and 3D detections, respectively. It then uses camera–LiDAR calibration and coordinate transforms to project the 3D detections onto the image plane as 2D boxes. Fusion focuses on the 2D coordinates from both sensors together with their confidence scores. An IoU-aware non-maximum suppression (NMS) associates and merges overlapped boxes to produce the final fused result. Therefore, the fusion decision relies on IoU between candidate boxes and their corresponding confidence scores.

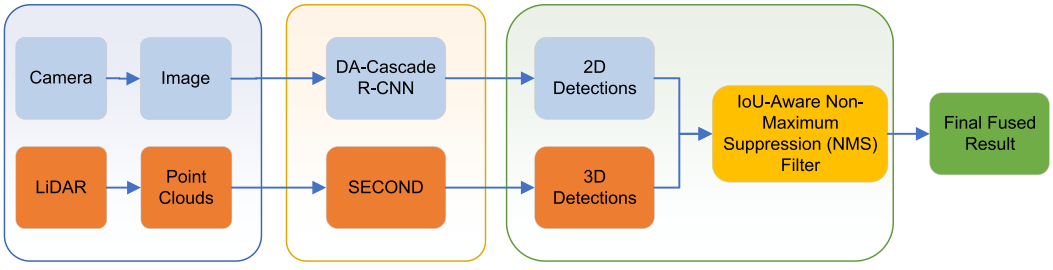


Figure 1: Framework of the proposed fusion object detection algorithm based on camera and LiDAR.

3.1 Dual Self-Attention Cascade R-CNN (DA-Cascade R-CNN)

Self-attention is an effective mechanism for modeling long-range dependencies by computing interactions among all spatial positions within a feature map. By reweighting features according to their global relevance, self-attention enables the network to emphasize informative regions while suppressing background noise. Recent studies have shown that incorporating attention mechanisms can significantly improve the detection of small or distant objects, and cascaded global attention has demonstrated effectiveness in remote sensing scenarios [21]. Building on these findings, we introduce Dual Self-Attention Cascade R-CNN (DA-Cascade R-CNN), which integrates self-attention into two complementary stages of the Cascade R-CNN pipeline: the backbone feature extraction stage and the region proposal network (RPN) stage. The term "Dual Self-Attention" explicitly refers to this two-stage design, where the same type of self-attention module is applied to both stages to enhance feature representation and proposal generation. To facilitate ablation analysis, the two stages with self-attention enabled individually are denoted as FE-Cascade R-CNN and RPN-Cascade R-CNN, respectively.

In DA-Cascade R-CNN, self-attention is implemented as a standalone module that can be seamlessly inserted into existing convolutional architectures. Each self-attention module follows a standard structure consisting of query, key, and value projections, followed by attention weight computation and feature aggregation. This design allows each spatial position to attend to all others, thereby injecting global contextual information into the feature maps.

To facilitate ablation analysis, we construct two Cascade R-CNN model variants by introducing self-attention into only one stage at a time. When self-attention is added solely to the feature-extraction stage, the resulting model is denoted as FE-Cascade R-CNN. When self-attention is introduced only at the RPN stage, the resulting model is denoted as RPN-Cascade R-CNN. Enabling self-attention at both stages yields the full DA-Cascade R-CNN model.

The details of each variant are described below.

(1) Feature-extraction stage with self-attention (FE-Cascade R-CNN): At the feature-extraction stage, four self-attention blocks are integrated into the backbone network after part of convolutional and ReLu layers. This configuration allows the backbone features to be globally refined before being forwarded to the region proposal network. By incorporating long-range contextual dependencies at an early stage, the self-attention blocks enhance salient object features while suppressing irrelevant background responses. Consequently, the feature representations provided to subsequent detection stages become more discriminative and context-aware. The structure of this stage is shown in Fig. 2.

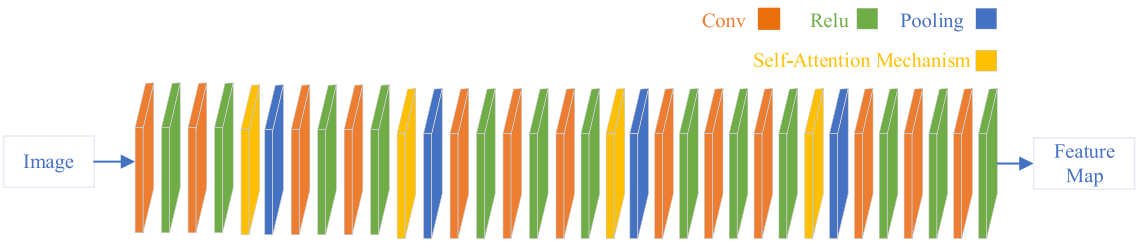


Figure 2: The backbone network with self-attention of FE-Cascade R-CNN.

(2) Region proposal network stage with self-attention (RPN-Cascade R-CNN): At the region proposal stage, a self-attention block is inserted after the convolutional layers of the region proposal network. This design enables proposal generation to benefit from global spatial context, improving the separation between foreground objects and background clutter. By leveraging long-range dependencies at the proposal stage, the

self-attention-enhanced RPN produces higher-quality candidate regions, particularly for small or sparsely distributed objects. The corresponding network structure of RPN in RPN-Cascade R-CNN is shown in Fig. 3.

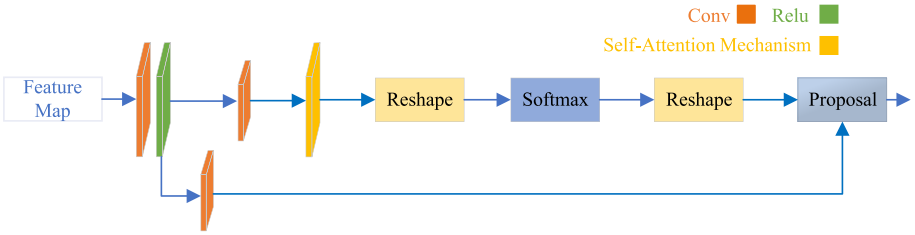


Figure 3: The region proposal network with self-attention of RPN-Cascade R-CNN.

(3) DA-Cascade R-CNN (dual self-attention integration): when self-attention is enabled at both the feature-extraction stage and the RPN stage, the resulting detector is referred to as DA-Cascade R-CNN. This dual self-attention design achieves an end-to-end enhancement from feature representation to proposal generation by combining the complementary strengths of FE-Cascade R-CNN and RPN-Cascade R-CNN. Compared with single-stage self-attention variants, the dual configuration better preserves informative features throughout the detection pipeline while progressively suppressing noise, leading to improved perception and representation across the entire detection process.

3.2 Fusion via IoU-Based Association and IoU-Aware NMS

Our fused method matches the detections from the camera with those from LiDAR using a deterministic one-to-one association rule. Any pair that satisfies the positive-match condition can enter the association and fusion step. The final fused output is then produced.

3.2.1 Sparse Tensor Construction

The model integrates detection candidates from different sensors by encoding them as a sparse input tensor. This yields a consistent, joint candidate set for the fusion network. The 2D camera detection candidates are represented as follows:

$$P^{2D} = \left\{P_1^{2D}, P_2^{2D}, \ldots, P_k^{2D}\right\}, \tag{1}$$

$$P_i^{2D} = \left\{\left[x_{i1}, y_{i1}, x_{i2}, y_{i2}\right], s_i^{2D}\right\}, \tag{2}$$

where P^{2D} denotes the set of k 2D detection candidates in a single image frame. $P_i^{2D}\ (i-1,2,\ldots,k)$ represents the i-th candidate object. The coordinates (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) specify the top-left and bottom-right corners of the 2D candidate's bounding box, respectively. The rectangle is determined by these two points. S_i^{2D} is the confidence score for the candidate detection box.

For point-cloud detection, the network outputs 3D candidate bounding boxes together with their localization attributes. The 3D detection candidate results are represented as follows:

$$P^{3D} = \left\{P_1^{3D}, P_2^{3D}, \ldots, P_n^{3D}\right\}, \tag{3}$$

$$P_i^{3D} = \left\{\left[h_i, w_i, l_i, x_i, y_i, z_i, \theta_i\right], s_i^{3D}\right\}, \tag{4}$$

where P^{3D} denotes the set of N 3D object detection candidate boxes from a LiDAR scan. $P_i^{3D}\ (j=1,2,\ldots,N)$ represents the j-th candidate detection box in a single LiDAR scan. The terms (h, w, l) are the dimensions

of the candidate box, (x, y, z) is its location, and θ is the yaw angle. S_i^{3D} represents the confidence score for the 3D object detection.

The essence of the algorithm for 2D and 3D detection candidates proposed in this paper is to filter and fuse the results from both sets of candidates. To avoid the problem where traditional single-sensor NMS might suppress correct detections, this paper first unifies the results from both sensors and then executes NMS to reduce false detections.

3.2.2 Association and Matching of Candidate Objects

In a given scene, object detection performed by cameras and LiDAR sensors, using their respective detection networks, yields a large number of candidate objects. Typically, the quantity of detection results from these two modalities is inconsistent, and their corresponding coordinates and confidence scores also differ. The proposed fusion process in this paper involves merging the two heterogeneous datasets into a single, unified dataset. Furthermore, detections corresponding to the same physical object must be consolidated. Therefore, it is necessary to associate and match the detection results from both sensor types.

The fusion process associates 2D object detections from camera images with 3D object detections from LiDAR point clouds. To accomplish this, the 3D candidate bounding boxes are first projected onto the 2D image plane. The degree of overlap between these projected 3D boxes and the original 2D boxes is then calculated. This overlap metric is used to match detections under the principle that a false positive is highly unlikely to share similar boundaries across both sensor modalities. By establishing this spatial correspondence, detections corresponding to the same object are effectively associated and unified.

Object matching between the two types of detectors is performed using the Intersection over Union (IoU) criterion, which quantifies the spatial overlap between two bounding boxes. The IoU is calculated as follows:

(1) Obtain the 2D candidate object candidates from the camera detector (as illustrated in region A of Fig. 4), and the corresponding detection confidence is denoted as P_i^{2D}. Each 2D detection result is represented by a rectangular bounding box parameterized by four values $(x_{min}, y_{min}, x_{max}, y_{max})$, which represent the bottom-left and top-right corners of the bounding box, respectively.

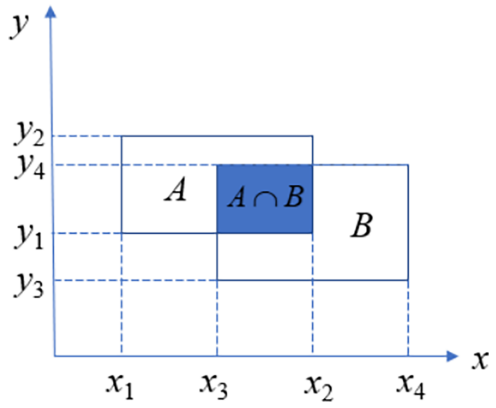


Figure 4: Diagram of bounding box overlap.

(2) Obtain the 3D candidate object results from the LiDAR-based detector with detection confidence P_i^{3D}. To enable spatial matching with 2D detections, the eight vertices of each 3D bounding box are projected onto the image plane through coordinate transformation, yielding a 2D bounding box consistent with the

2D detection format (as shown in region B of Fig. 4). The core projection from LiDAR coordinates to the image plane is formulated as follows:

$$y = P_{rect}^{(i)} R_{rect}^{(0)} T_{velo}^{cam} X, \tag{5}$$

where X represents the point cloud data in homogeneous coordinates. The matrix $P_{rect}^{(i)}$ is derived from P_{rect}^{0i} in the KITTI's calibration file calib_cam_to_cam.txt, while $R_{rect}^{(0)}$ is obtained by extending the 3 × 3 rectification matrix P_{rect}^{00} from calib_cam_to_cam.txt into a 4 × 4 form. The transformation matrix T_{Velo}^{cam}, expanded from the rotation and translation parameters provided in calib_velo_to_cam.txt, is defined below:

$$T_{velo}^{cam} = \begin{pmatrix} R_{velo}^{cam} & t_{velo}^{cam} \\ 0 & 1 \end{pmatrix}. \tag{6}$$

A schematic illustration of the overlap between the projected 3D bounding box and the 2D detection box is shown in Fig. 4.

(3) The IoU value is then computed as the ratio between the area of intersection and the area of union of the two 2D bounding boxes:

$$IOU = \frac{A \cap B}{A \cup B}. \tag{7}$$

By definition, IoU values range from 0 to 1, where an IoU of 0 indicates no spatial overlap between the two bounding boxes, and larger values correspond to stronger spatial association. IoU is subsequently used as the criterion for matching and fusion between camera-based and LiDAR-based detection results.

3.2.3 Data Filtering Based on IoU and Confidence

After calculating the overlap ratio based on IoU, one or more detection results may exist for the same object that is detected by both the camera and LiDAR. Multiple fusion results persist even after the IoU association is made. Therefore, to ensure accurate detection, duplicate detection boxes are filtered.

Among current object detection algorithms, the most prominent technique for removing redundant candidate detections is the NMS algorithm. Based on NMS logic, the following operations and adjustments are performed:

(1) Sort detections in descending order based on confidence scores.
(2) Prioritize the detection with the highest confidence score, then filter related object detections using an IoU threshold.

This algorithm presents the following issues:

(1) During the initial sorting step, prioritizing confidence scores may result in high-confidence detections with low IoU appearing first. This leads to filtering out high-confidence, low-IoU detections while allowing low-confidence detections to pass.
(2) Results with high IoU but low confidence scores are ranked after those with low IoU but high confidence scores, reducing the Average Precision (AP) of IoU-based detection results.

To address these issues, the NMS optimization scheme proposed in [22] is adopted. By incorporating IoU into the NMS framework, it mitigates the aforementioned problems to a certain extent and improves the quality of object screening.

4 Experiment and Analysis

To objectively and quantitatively evaluate the effectiveness of the camera–LiDAR fusion algorithm proposed in this article, this section presents a series of experiments for validation and analysis. These experiments are conducted on the public KITTI dataset, comparing the detection results of our algorithm with those of baseline models and other advanced methods to verify its performance in scenarios of varying difficulty. To ensure a scientific and rigorous evaluation, it is first necessary to define a standard set of performance evaluation metrics.

4.1 Evaluation Metrics

Object detection performance is commonly evaluated using Recall, Precision, and Average Precision (AP), which jointly reflect a detector's ability to identify objects accurately and comprehensively. Recall measures the proportion of ground-truth objects that are correctly detected, indicating the model's ability to find relevant objects. It is defined as:

$$Recall = \frac{TP}{TP + FN}, \tag{8}$$

where TP (true positives) represents the number of correctly detected objects, and FN (false negatives) denotes the number of ground-truth objects that are missed. Precision evaluates the proportion of correctly detected objects among all detections, reflecting detection accuracy. It is computed as:

$$Precision = \frac{TP}{TP + FP}, \tag{9}$$

where FP (false positives) refers to non-target objects being incorrectly detected.

In the field of object detection, Intersection over Union (IoU) is commonly used to determine TP and FP. A predicted bounding box is considered a true positive if its IoU with the corresponding ground-truth box exceeds a predefined threshold; otherwise, it is treated as a false positive. Since recall and precision often exhibit a trade-off relationship, Average Precision (AP) is widely adopted as a comprehensive evaluation metric. AP summarizes detection performance by computing the area under the Precision–Recall (PR) curve across different recall levels. It is defined as follows:

$$AP = \int_0^1 p\left(r\right) dr, \tag{10}$$

where r denotes recall, and $p(r)$ represents the precision corresponding to recall level r on the Precision–Recall curve. In practice, the PR curve is obtained from a set of discrete detection results ranked by confidence scores, and AP provides an overall measure of detection performance across varying operating points.

4.2 Experimental Results and Comparisons

In the KITTI dataset, only the training samples are provided with ground-truth labels, while the labels of the test samples are not publicly available and must be evaluated throughthe official KITTI server. Accordingly, the 7481 labeled training samples are randomly divided into 3769 samples for training and 3712 samples for testing in our experiments. By evaluating the method's effectiveness across different levels of scenes, the AP for 3D object detection is assessed. Scene difficulty classification is closely related to occlusion and clipping phenomena, which necessitate a detailed presentation of both IoU and two distinct threshold selections during precision calculations, as shown in Table 1. The terms "Easy," "Moderate," and "Hard" in the table denote the difficulty levels of object detection scenarios, while the precision calculations for different

IoU thresholds are represented by 0.7 and 0.5, respectively. Specifically, a detection result is deemed correct when the IoU between the predicted bounding box and the actual object exceeds the IoU threshold.

Table 1: Average precision comparison between Cascade R-CNN variants and the original algorithm (%).

	Easy	Moderate	Hard
Cascade R-CNN	90.68	89.95	78.40
FE-Cascade R-CNN	91.71	90.63	81.72
RPN-Cascade R-CNN	90.47	89.74	80.46
DA-Cascade R-CNN	92.13	91.25	83.22

As shown in Table 1, the four variants demonstrate outstanding performance across the three difficulty levels in terms of object detection average precision. Specifically:

Cascade R-CNN achieves an average precision of 90.68% at the Easy level, 89.95% at the moderate level, and 78.40% at the hard level. This indicates that Cascade R-CNN demonstrates high effectiveness in handling moderately complex object detection tasks, but its performance slightly declines when processing the most challenging scenarios.

The feature extraction-optimized Cascade R-CNN (FE-Cascade R-CNN) achieved average precisions of 91.71%, 90.63%, and 81.72% across the three difficulty levels, as shown in Table 1, outperforming the original Cascade R-CNN. This indicates that feature extraction optimization demonstrates a slight advantage across all difficulty levels, particularly in handling the most challenging scenarios. This suggests that the feature extraction stage enhances feature capture capabilities, preserving more effective features.

The Cascade R-CNN optimized at the RPN stage (RPN-Cascade R-CNN) achieved an average precision of 90.47% at the easy level, 89.74% at the moderate level, and 80.46% at the hard level. Compared to Cascade R-CNN + FE-Self, this optimization method exhibits slightly lower AP across all difficulty levels, indicating moderate performance in object detection. The primary reason lies in the deep learning process, which also discards ineffective features during feature extraction, thereby compromising the quality of candidate region proposals and ultimately impacting the final AP rate.

The dual attention mechanism (DA-Cascade R-CNN) simultaneously optimizes feature extraction and the RPN, combining the strengths of both components. This approach enhances feature extraction capabilities while improving the quality of region proposals, achieving AP of 92.13%, 91.25%, and 83.22% across three difficulty levels. Among all proposed solutions, this simultaneous optimization demonstrates the most outstanding performance, particularly exhibiting a significant advantage when handling scenes at the hard difficulty level.

In summary, across the three difficulty levels of object detection tasks, DA-Cascade R-CNN demonstrated the highest AP, indicating the strongest generalization capability and optimal performance. The results are compared in Fig. 5 below: (a) shows the Cascade R-CNN detection results; the object in the upper left corner and the heavily occluded object with low light intensity in the center were not detected. Additionally, the confidence score for the occluded object in the middle right was low. While false positives appear in the lower left corner. (b) shows FE-Cascade R-CNN results: confidence for the proper occluded object improves, but detection fails for the small upper-left object and the dimly lit small object in the center; the false positive on the left trash can is eliminated. (c) presents RPN-Cascade R-CNN results: AP decreases compared to feature network optimization, failing to detect the small object in the upper left corner and the object in the center. (d) DA-Cascade R-CNN shows significantly improved detection of small objects in the

upper left corner. While detection AP is slightly higher than FE-Cascade R-CNN, the small object in the center and the occluded object in the upper left corner remain undetectable.

(a) Cascade R-CNN **(b)** FE-Cascade R-CNN

(c) RPN-Cascade R-CNN **(d)** DA-Cascade R-CNN

Figure 5: Examples of detection results from different variants of Cascade R-CNN.

4.3 Verification of Fused Camera and LiDAR

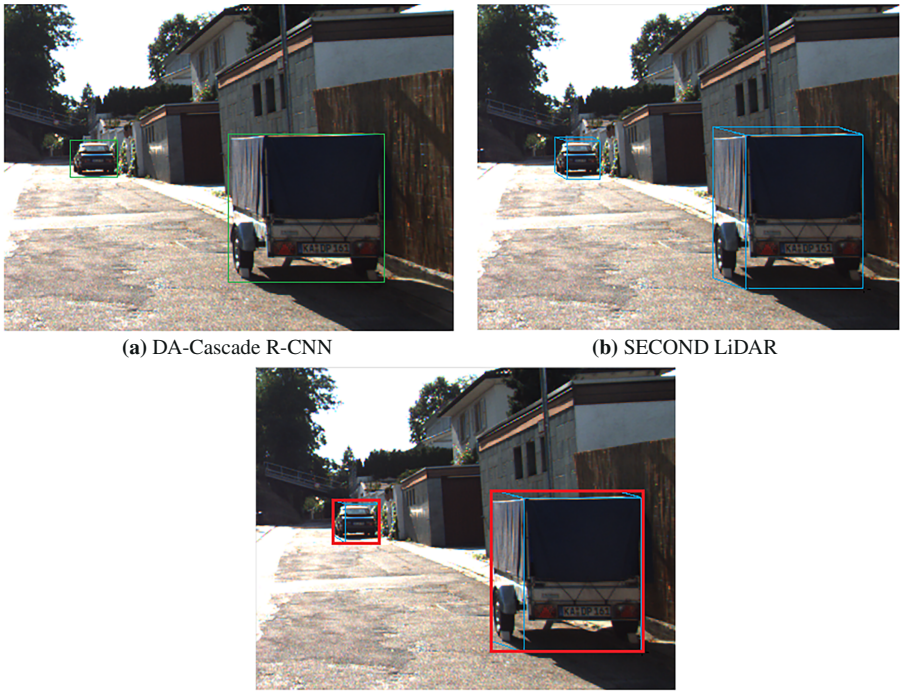
The camera image detection results obtained using the selected camera detection algorithms, DA-Cascade R-CNN, SECOND, and the fusion detection results are shown in Fig. 6.

As shown in Fig. 6a, the detection results of the DA-Cascade R-CNN algorithm cannot only accurately identify the large trailer target near the right side of the picture, but also can generate 2D detection boxes close to the edge for the smaller vehicle target on the left side in the distance; Fig. 6b shows the detection results of the LiDAR point cloud target detection algorithm SECOND. After projecting the generated 3D detection boxes onto the image, it can be seen that the radar algorithm can provide precise target spatial depth information; Fig. 6c shows the final fusion result of the camera and LiDAR. By combining the visual semantic advantages of (a) and the spatial position advantages of (b), after calculating the IoU and filtering with NMS, redundant detection boxes are effectively eliminated, and high-confidence targets are retained. The detection results of the single sensor and the final fusion result both show high average precision, and the fusion result further improves the reliability of the detection box while ensuring 3D information, proving the effectiveness of the static fusion strategy.

As shown in the figure above, the detection results from fused cameras and LiDAR data exhibit high average precision. To better highlight the advantages of fusion over single sensors, detection comparisons were conducted on the KITTI dataset. The comparison algorithms include Faster R-CNN, YOLOv5 [17], MS-CNN [23], Cascade R-CNN, and our proposed several improved and fused algorithms. Detection average precision statistics under three detection difficulty levels (easy, moderate, hard) are presented in Table 2, with an IoU threshold of 0.5.

As shown in Table 2, the baseline Cascade R-CNN algorithm demonstrates outstanding performance among all single-object detection methods, achieving higher detection rates than other algorithms across all three difficulty levels. Compared to Faster R-CNN, it outperforms by 5.87%, 3.77%, and 0.37%, respectively. It also outperforms YOLOv5 by 2.51%, 11.25%, and 8.95% in average precision, respectively. Compared to MS-CNN, it achieves higher detection rates by 0.46%, 0.87%, and 1.9%. When combining Cascade R-CNN with the SECOND algorithm, the results surpass those of the aforementioned single-sensor detection methods. Notably, it outperforms the original Cascade R-CNN by 2.84%, 1.72%, and 6.96% in each of the respective metrics. The data indicate that the fusion algorithm significantly improves detection AP across all

difficulty levels. This fusion strategy effectively compensates for the limitations of single detection methods. In this fusion algorithm, replacing the original Cascade R-CNN with DA-Cascade R-CNN demonstrates that improvements in single-sensor detection AP indirectly enhance fusion detection AP. Compared to the initial algorithm, this replacement achieves increases of 1.6%, 0.7%, and 4.05%, respectively. This effectively validates that enhancing the detection AP of individual sensors can improve the overall fusion AP of the combined algorithm.



(a) DA-Cascade R-CNN **(b)** SECOND LiDAR

(c) Our static fusion method

Figure 6: Examples of single sensor and fusion detection results.

Table 2: 2D object detection average precisions statistics (%).

	Easy (0.5)	**Moderate (0.5)**	**Hard (0.5)**
Faster R-CNN	84.81	86.18	78.01
YOLOv5 [22]	88.17	78.70	69.45
MS-CNN [23]	90.22	89.08	76.50
Cascade R-CNN	90.68	89.95	78.40
FE-Cascade R-CNN	91.71	90.63	81.72
RPN-Cascade R-CNN	90.47	89.74	80.46
DA-Cascade R-CNN	92.13	91.25	83.22
Fusion with Cascade R-CNN	93.52	91.67	85.36
Fusion with DA-Cascade R-CNN	95.12	92.37	89.41

5 Conclusion

This paper presents an effective fusion algorithm that combines 2D image-based and 3D LiDAR-based detection candidates by leveraging Intersection over Union (IoU) and confidence scores as the primary criteria for fusion. The proposed method enhances object detection performance by incorporating self-attention mechanisms into the Cascade R-CNN architecture, which improves the detector's ability to perceive small and distant objects. Experimental results demonstrate consistent gains in both recognition and localization average precision across various levels of task difficulty. However, the method also has inherent limitations. Specifically, its decision-level association is sensitive to cross-modal extrinsic calibration and confidence score inconsistencies. Furthermore, static fusion may struggle in dynamic environments with changing conditions.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Jianping Wu: methodology, software, experiments, data curation, visualization. Nian Li: writing—original draft and editing. Libin Dong and Ping Zhang: supervision and critical review. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this study come from the KITTI Vision Benchmark Suite (Geiger et al., CVPR 2012) and are openly available under the KITTI research license at https://www.cvlibs.net/datasets/kitti/.

Ethics Approval: Not applicable.

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

References

1. Zou Z, Chen K, Shi Z, Guo Y, Ye J. Object detection in 20 years: a survey. Proc IEEE. 2023;111(3):257–76. doi:10.1109/jproc.2023.3238524.
2. Cao Z, Zhang H, Liang L, Wang H, Jin S, Ye Li G. Task-oriented semantic communication for stereo-vision 3D object detection. IEEE Trans Commun. 2025;73(9):7552–67. doi:10.1109/TCOMM.2025.3545687.
3. Sindagi VA, Zhou Y, Tuzel O. MVX-net: multimodal VoxelNet for 3D object detection. In: Proceedings of the 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. p. 7276–82. doi:10.1109/icra.2019.8794195.
4. Cao P, Chen H, Zhang Y, Wang G. Multi-view frustum pointnet for object detection in autonomous driving. In: Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP); 2019 Sep 22–25; Taipei, Taiwan. p. 3896–9. doi:10.1109/icip.2019.8803572.
5. Solovyev R, Wang W, Gabruseva T. Weighted boxes fusion: ensembling boxes from different object detection models. Image Vis Comput. 2021;107:104117. doi:10.1016/j.imavis.2021.104117.
6. Neubeck A, Van Gool L. Efficient non-maximum suppression. In: Proceedings of the 18th International Conference on Pattern Recognition; 2006 Aug 20–24; Hong Kong, China. p. 850–5. doi:10.1109/ICPR.2006.479.
7. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
8. Ali ML, Zhang Z. The YOLO framework: a comprehensive review of evolution, applications, and benchmarks in object detection. Computers. 2024;13(12):336. doi:10.3390/computers13120336.
9. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Computer vision—ECCV 2016. Berlin/Heidelberg, Germany: Springer; 2016. p. 21–37. doi:10.1007/978-3-319-46448-0_2.

10. Bai X, Hu Z, Zhu X, Huang Q, Chen Y, Fu H, et al. TransFusion: robust LiDAR-camera fusion for 3D object detection with transformers. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 1080–9. doi:10.1109/CVPR52688.2022.00116.
11. Zhao H, Guan R, Wu T, Man KL, Yu L, Yue Y. UniBEVFusion: unified radar-vision bevfusion for 3D object detection. In: Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA); 2025 May 19–23; Atlanta, GA, USA. p. 6321–7. doi:10.1109/ICRA55743.2025.11128067.
12. Choi J, Shin M, Paik J. Fusion of an RGB camera and LiDAR sensor through a Graph CNN for 3D object detection. Opt Continuum. 2023;2(5):1166. doi:10.1364/optcon.479777.
13. Cai Z, Vasconcelos N. Cascade R-CNN: delving into high quality object detection. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 6154–62. doi:10.1109/CVPR.2018.00644.
14. Bodla N, Singh B, Chellappa R, Davis LS. Soft-NMS—Improving object detection with one line of code. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 5562–70. doi:10.1109/iccv.2017.593.
15. Yan Y, Mao Y, Li B. SECOND: sparsely embedded convolutional detection. Sensors. 2018;18(10):3337. doi:10.3390/s18103337.
16. Liu Y, Meng S, Wang H, Liu J. Deep learning based object detection from multi-modal sensors: an overview. Multimed Tools Appl. 2024;83(7):19841–70. doi:10.1007/s11042-023-16275-z.
17. Ultralytics. YOLOv5 [Internet]. San Francisco, CA, USA: GitHub; 2020 [cited 2026 Jan 1]. Available from: https://github.com/ultralytics/yolov5.
18. Shi C, Wang C, Sun S, Liu X, Xi G, Ding Y. LiDAR point cloud object recognition method via intensity image compensation. Electronics. 2023;12(9):2087. doi:10.3390/electronics12092087.
19. Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 4490–9. doi:10.1109/CVPR.2018.00472.
20. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012 Jun 16–21; Providence, RI, USA. p. 3354–61. doi:10.1109/CVPR.2012.6248074.
21. Yang Z, Liu Y, Wen G, Xia X, Zhang WE, Chen T. Object detection in remote sensing images with parallel feature fusion and cascade global attention head. IEEE Geosci Remote Sens Lett. 2024;21:6007205. doi:10.1109/LGRS.2024.3385231.
22. Wu S, Li X, Wang X. IoU-aware single-stage object detector for accurate localization. Image Vis Comput. 2020;97:103911. doi:10.1016/j.imavis.2020.103911.
23. Cai Z, Fan Q, Feris RS, Vasconcelos N. A unified multi-scale deep convolutional neural network for fast object detection. In: Computer Vision—ECCV 2016. Berlin/Heidelberg, Germany: Springer; 2016. p. 354–70. doi:10.1007/978-3-319-46493-0_22.