



ARTICLE

Hybrid Embedded System for Identifying Road Condition Defects Using Three Adapted Deep Learning Models

Feng-Cheng Lin^{*}, Jia-Yan Lin, Chun-Yu Hung and Wijaya Wijaya

Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan

^{*}Corresponding Author: Feng-Cheng Lin. Email: fc.lin@fcu.edu.tw

Received: 05 May 2026; Accepted: 01 September 2026; Published: 22 October 2026

ABSTRACT: Road defects, such as cracks and potholes, pose significant challenges to urban infrastructure management and transportation safety. Existing automated detection methods often suffer from computational inefficiency, sensitivity to environmental conditions, and a lack of severity assessment capabilities. Addressing these limitations, this study presents an integrated deep learning framework optimized for embedded platforms, designed to provide comprehensive, real-time road defect detection, classification, and tracking. The proposed framework utilizes the proposed FAST-UNet for efficient segmentation, an enhanced YOLOv11-SDIDC for robust detection, MobileNetV3 with SimAM attention for multi-level severity classification, and ByteTrack for stable tracking, optimized with a hybrid loss function. Deployed on the NVIDIA Jetson AGX Orin platform using TensorRT acceleration, the system demonstrates competitive predictive performance and efficient inference on the RDD2022 dataset. Furthermore, integrated GPS functionality enables real-time geospatial mapping. This research contributes a scalable, efficient, and multi-functional solution for automated road defect management, enhancing intelligent transportation systems in smart cities.

KEYWORDS: Deep learning; road defect detection; MobileNetV3; YOLOv11-SDIDC; ByteTrack

1 Introduction

Well-maintained road infrastructure is vital for urban development, boosting transport efficiency and public safety. However, persistent stress from heavy traffic, variable weather, and inadequate maintenance leads to numerous defects like potholes, cracks, and joint failures. These issues reduce driving comfort and increase accident risks, posing significant challenges for municipalities needing to perform timely, cost-effective maintenance. Traditional manual inspections are labor-intensive, prone to error, and impractical for large-scale assessments [1]. Fortunately, deep learning advancements over the last decade have paved the way for automated road defect detection, offering more robust and precise methods for identifying infrastructure problems.

Among these deep learning approaches, object detection and segmentation models have proven particularly effective for identifying road surface defects. Initial efforts often involved convolutional neural networks focused on single defect types, primarily potholes, due to their significant impact on vehicle safety and road conditions. While state-of-the-art detectors like YOLOv4 [2], YOLOv5 [3], and YOLOv7 [4] demonstrate promising real-time capabilities, this single-class focus becomes a limitation when faced with real-world conditions. Road surfaces present diverse issues—including various crack types (transverse, longitudinal, alligator) and expansion joint misalignments—which single-class methods struggle to handle

comprehensively. Consequently, there is a clear need for unified frameworks capable of simultaneously detecting multiple defect categories and providing detailed segmentation of their boundaries.

However, developing and training such sophisticated models is hampered by another significant challenge: the lack of standardized, comprehensive datasets. For example, while RDD2022 [5] aggregated annotated images internationally, its annotations primarily cover cracks and potholes, overlooking other structural elements. Similarly, specialized datasets, such as those created for cracks under shadow occlusion [6], often lack thorough documentation of crucial objects like manholes or speed bumps. Even alternative data acquisition methods, like UAV-based detection [1], present trade-offs; while offering wider coverage, high-altitude aerial views can miss subtle pavement details. These limitations underscore the critical need for diverse datasets that capture multiple defect types under varied environmental conditions (lighting, weather, wear patterns) to ensure the development of truly robust and generalizable models.

While addressing data limitations is crucial, advancements on the modeling front have also been substantial. Recent investigations highlight the value of integrated architectures and specialized techniques tailored for defect analysis. Key developments include multi-scale feature networks [7] for improving detail resolution in cracks, efficient backbones like MobileNetV3 [8] optimized for resource-constrained settings, and attention mechanisms like SimAM [9] for refining features with minimal overhead. Furthermore, techniques like Focal Loss [10] help mitigate label imbalance inherent in defect data (compared to standard Cross-Entropy [11]), ensuring focus on rare yet critical defects, while tracking methods like ByteTrack [12] enable the monitoring of defects over time. Despite these powerful individual tools, their comprehensive integration into single, end-to-end pipelines remains uncommon. This integration gap makes it difficult for municipalities to deploy the large-scale, real-time, multi-functional inspection systems required to meet complex operational needs.

To bridge this integration gap, we distinguish our system from closely related real-time edge frameworks, such as the embedded detection and localization system. While that pioneer study demonstrated real-time embedded detection and mapping, our framework advances edge infrastructure monitoring through a co-designed, four-stage integrated pipeline.

Rather than introducing a singular un-optimized model backbone, our system-level novelty lies in the execution of (i) a four-stage cascading pipeline combining road-region isolation (FAST-UNet), multi-class target detection, three-level severity categorization, and frame-to-frame temporal tracking, (ii) a substantial multi-task dataset expansion adding structural road components and a dedicated 4270-sample distress severity dataset, and (iii) an extremely efficient edge-computing footprint utilizing a 1.99M-parameter detector and a 0.31M-parameter segmenter optimized for highly resource-constrained transportation edge nodes.

2 Related Works

2.1 Detection of Pavement Distresses Using UAV Imagery and YOLO-Based Methods

As explored in recent studies, Unmanned Aerial Vehicle (UAV) imagery offers valuable high-resolution, top-down perspectives for road inspection. Early work adapted models such as YOLOv4 [2] and YOLOv5 [3] to identify cracks and potholes despite varied lighting and complex backgrounds. Building on this, Zou et al. [6] improved detection accuracy, particularly for challenging scenarios like heavy shadows and textured surfaces, by integrating advanced components (like SPPCSPC_G, GELU activation, and K-means++ anchor refinement) into YOLOv7. Although these UAV-centric approaches bring significant benefits in reduced inspection time and manpower, their application has concentrated mainly on common defects like major cracks and potholes. This often leaves other anomalies, such as expansion joints and manholes,

undetected, underscoring the need for systems capable of robust performance in varied conditions and more comprehensive defect coverage.

2.2 Domain Adaptation and Multi-Scale Feature Extraction

Research also focuses on improving model robustness, addressing challenges like regional variations and feature detail. To handle regional variations, frameworks such as Domain Adaptive Road Damage Detection (DA-RDD) [13] employ techniques like adversarial adaptation and CycleGAN image translation alongside Faster R-CNN [14] to align cross-domain feature distributions. While effective for generalization, these approaches often overlook multi-class detection needs, typically focusing only on cracks and potholes. Complementing domain adaptation efforts, methods aiming to enhance feature representation, like the Multi-Scale Feature Attention Network (MDA) [7], use dilated convolutions and attention to fuse multi-scale context. This balances localization performance but significantly increases parameter counts, creating an unfavorable latency trade-off on edge architectures. To systematically evaluate these paradigms, our approach is explicitly benchmarked against domain-specific state-of-the-art architectures, including the Domain Adaptive Road Damage Detection (DA-RDD) framework and the Multi-Scale Feature Attention Network (MDA). While DA-RDD focuses primarily on cross-domain distribution alignments for basic cracks/potholes and MDA handles feature detail via dense dilated convolutions, both methods lack structural component awareness and real-time tracking integration. A comprehensive architectural comparison between our framework and the global-scale embedded inspection baseline is detailed in [Table 1](#).

Table 1: Cross-study comparison of embedded road monitoring frameworks.

System/Study Attribute	Global-Scale Embedded System Baseline	Proposed Hybrid Embedded Framework (This Work)
Dataset Scale/Source	Global Crowdsourced RDD Images	Expanded RDD2022 + Multi-Class Structural Extensions
Defect Classes Covered	Standard Cracks, Basic Potholes	Cracks, Potholes, Speed Bumps, Manholes, Bridge Expansion Joints
Severity Assessment	Binary/Present-Absent Only	Three-Level Pavement Engineering Severity Grading
Localization Approach	API/Smartphone GPS Logging	Integrated Hardware NEO-M10 GPS via GPIO Synchronization
Real-Road Validation	Fleet Crowdsourcing Profiles	Controlled Real-Vehicle Edge Runs with Vibration & Thermal Analysis

2.3 Lightweight Networks for Embedded Deployment

Given the computational constraints of embedded devices, lightweight models have become essential for practical road inspection applications. Recent research leverages architectures like MobileNetV3, which employs inverted residual blocks and Squeeze-and-Excitation (SE) modules to enhance feature discrimination without significantly increasing complexity [8]. To further improve efficiency and focus, attention mechanisms such as SimAM have been integrated into these networks, highlighting critical defect regions while maintaining a low parameter count suitable for real-time performance on resource-constrained platforms [9]. Complementary lightweight segmentation architectures, like Lite-SRGAN and Lite-UNet [15],

contribute to efficient road surface isolation by separating relevant regions from background elements (e.g., curbs and vegetation), thereby reducing the computational burden for subsequent detection tasks.

While these lightweight approaches have yielded commendable improvements in speed and efficiency [15], significant limitations remain, particularly in providing detailed, actionable defect analysis. For instance, classification tasks often struggle with precise localization, and detection methods may fail to capture the complete geometric attributes of identified defects. Moreover, the common focus on semantic segmentation overlooks the critical need for individualized analysis of each defect instance. Ideally, an automated system should provide precise localization and contour information for each distress occurrence, enabling maintenance personnel to accurately understand the specific location and shape of every defect.

This gap between current capabilities and practical needs often stems from fragmented approaches. Most existing systems treat detection, classification, and segmentation as separate tasks rather than integrated framework components. Consequently, they lack the ability to simultaneously address these challenges—along with dynamic tracking—in real time on embedded platforms such as the NVIDIA Jetson AGX Orin. This lack of integration, coupled with the inadequacy of traditional manual inspections for modern demands, underscores the need for more advanced, unified automated solutions.

Therefore, our work is motivated by the need to address these specific limitations. By synthesizing the advantages of lightweight architectures and advanced attention mechanisms, we aim to develop an integrated, multi-task system. This system is designed not only to enhance detection and segmentation performance but also to provide the detailed, instance-level analysis crucial for effective road maintenance. Our unified solution targets the high precision, efficiency, and real-time inference capabilities essential for modern infrastructure management.

2.4 RDD2022 Dataset

Alongside methodological advances, large-scale annotated datasets have emerged. RDD2022, for instance, is prominent for compiling diverse images from multiple countries under varied road and weather conditions [5]. It provides bounding box annotations for common defects like potholes and various crack types (alligator, transverse, longitudinal). This inherent diversity helps mitigate overfitting to specific regions or climates, boosting model generalizability. However, RDD2022 primarily covers the types of damage illustrated, lacking comprehensive labels for other structural anomalies such as speed bumps, expansion joints, and manholes. Researchers often address this gap by supplementing RDD2022 with additional task-specific annotations. This strategy allows them to leverage RDD2022's broad diversity—including varied road textures (asphalt vs. concrete), illumination scenarios (cloudy vs. sunny), and traffic conditions—while incorporating more specialized anomalies, proving particularly beneficial for training robust multi-class detection models.

3 Materials and Methods

3.1 The Overall Architecture of the Proposed Method

The proposed system, illustrated in Fig. 1, offers an integrated deep learning framework for comprehensive road defect analysis and optimized for real-world deployment on embedded platforms. It operates via a four-stage sequential pipeline: (1) Road segmentation using FAST-UNet isolates the road area; (2) Defect detection with the YOLOv11_SDIDC [16] model identifies potential anomalies; (3) Severity classification employs an Improved-MobileNetV3 integrated with the SimAM attention mechanism [8,9]; and (4) Defect tracking uses the ByteTrack algorithm [12]. Each stage feeds its output to the next, creating an efficient process that enhances accuracy. This integrated, multi-stage design overcomes single-task model limitations by

comprehensively managing multi-class detection, classification, segmentation, and tracking within dynamic environments, offering a robust and practical monitoring solution. Further details on each stage are provided in [Sections 3.2–3.6](#).

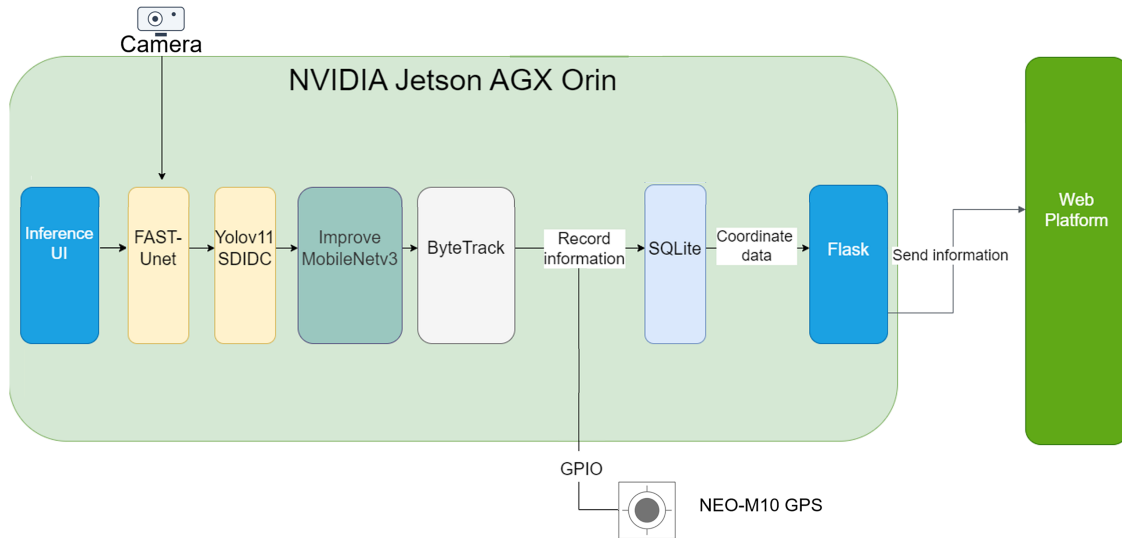


Figure 1: Proposed system architecture diagram.

3.2 Road Segmentation

The proposed FAST-UNet model, the architecture of which is detailed in [Fig. 2](#), efficiently segments road areas to enhance subsequent defect detection accuracy.

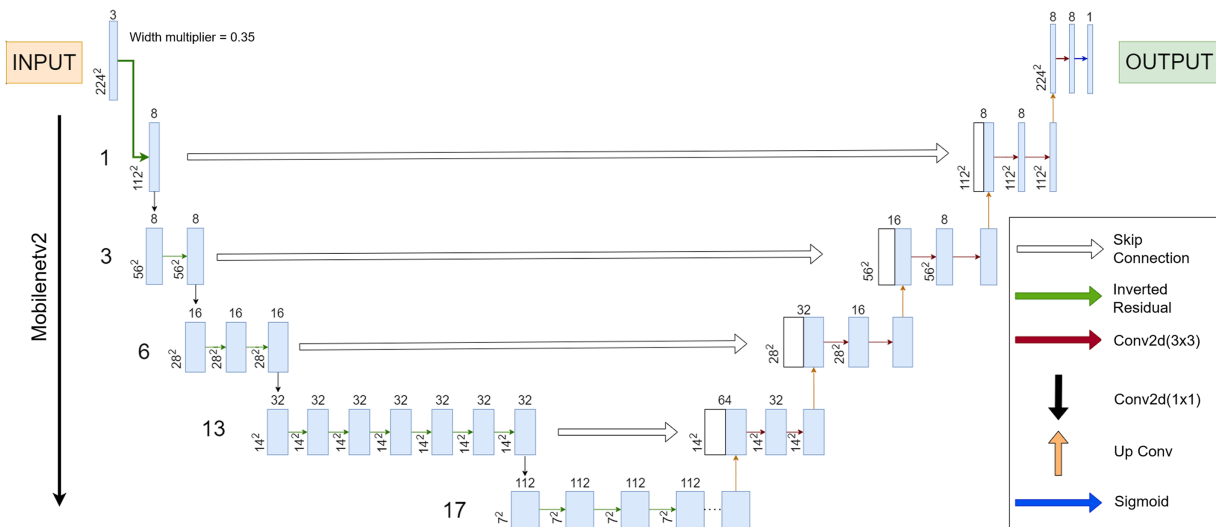


Figure 2: The proposed FAST-UNet architecture diagram.

Based on a lightweight U-Net architecture [15] and UNet [17], it is optimized for computational efficiency and embedded platforms such as the NVIDIA Jetson AGX Orin [18]. The integration of MobileNetV2 inverted residual blocks [19] and Squeeze-and-Excitation (SE) attention modules [20] boosts segmentation accuracy with minimal computational overhead. Robustness was achieved through training on carefully

annotated images from diverse scenarios within the RDD2022 dataset, covering intersections, varied textures, and occlusions. Additionally, data augmentation techniques like rotation, scaling, and brightness adjustments were systematically used to improve generalization across diverse environments.

3.3 Defect Detection

The proposed YOLOv11_SDIDC model (Fig. 3) modifies the original YOLOv11 structure. This new model was developed for robust multi-category road defect identification under diverse and challenging conditions. It integrates SCDwn modules for improved context preservation and InceptionNext blocks for comprehensive multi-scale feature extraction [21], enabling efficient detection of small, occluded, or ambiguous defects. Trained on the extensive and varied RDD2022 dataset, the model benefits from enhanced robustness and adaptability. Advanced data augmentation techniques—including mosaic augmentation, random flipping, color jittering, and multi-scale training—were extensively employed. Furthermore, integrating Focal Loss [10] to handle class imbalance and Complete Intersection-over-Union (CIoU) Loss [22] for precise bounding box refinement significantly boosted detection accuracy compared to conventional YOLO architectures.

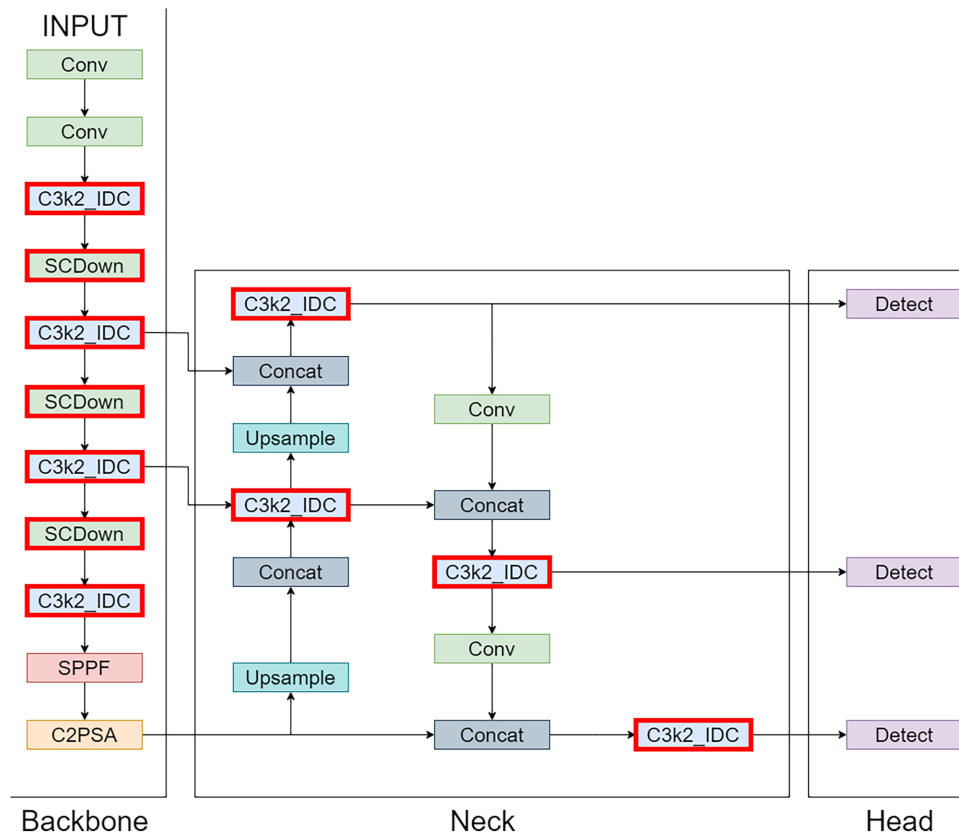


Figure 3: The proposed YOLOv11_SDIDC architecture.

3.4 Severity Classification

For effective severity classification (minor, moderate, severe), the proposed system utilizes an improved MobileNetV3 model synergistically coupled with the SimAM attention mechanism [9]. Its computational efficiency, suitable for embedded deployment, stems from MobileNetV3's refined bottleneck structure [8], which incorporates the inverted residual with squeeze-and-excitation design (illustrated in Fig. 4). While this

architecture ensures efficiency, the parameter-free SimAM [9] enhances feature representation by emphasizing crucial defect characteristics without added overhead, allowing for accurate severity assessment. The classification model was trained on a dataset meticulously created by cropping defects from RDD2022 images and labeling them based on maintenance standards. Strategic use of hard sample mining and balanced augmentation further enhanced model robustness and classification accuracy across varying severity levels.

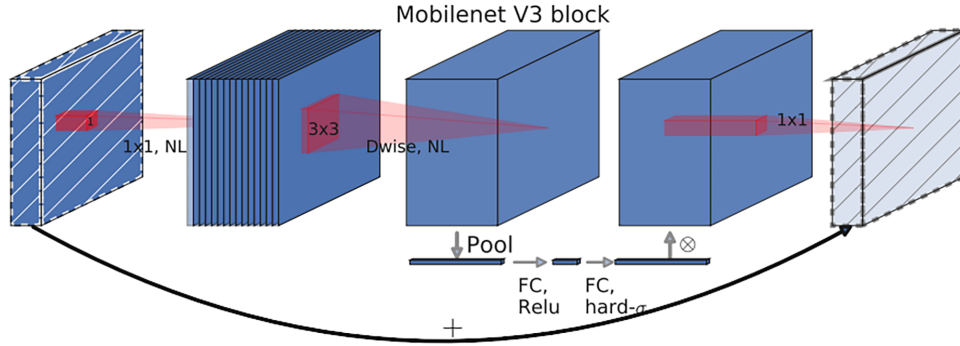


Figure 4: Inverted residuals with squeeze-and-excitation [8].

3.5 Defect Tracking

Defect tracking across sequential video frames is managed via the ByteTrack algorithm [12]. Although physical road defects are static structural features, they are captured via a forward-moving vehicular platform. Egomotion creates dynamic scale variations, structural foreshortening, and transient lens occlusions across consecutive frames, causing standard non-tracking detectors to register multiple duplicate counts of a single defect.

Incorporating ByteTrack introduces cross-frame temporal association, assigning an invariant tracking ID to each unique defect occurrence, as illustrated in Fig. 5. This multi-frame verification filters out spurious false positives and forms the mathematical foundation for down-pipeline GPS-based deduplication during spatial coordinate logging.

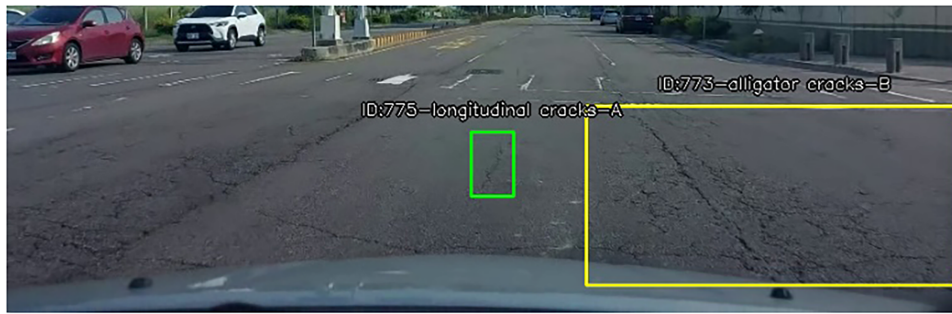


Figure 5: ByteTrack assigning unique tracking IDs to distinct road surface defect instances across sequential frames (Source: Drawn in this research).

3.6 Loss Functions and Attention Mechanism

The system employs specific hybrid loss functions to optimize performance across its different tasks. Combining Dice Loss [17] with Binary Cross-Entropy Loss enhances road boundary segmentation accuracy for precise delineation. Focal Loss [10] addresses significant class imbalance for defect detection, while CIoU

Loss [22] improves bounding box refinement. Severity classification relies on the standard Cross-Entropy Loss [11] which is defined in Eq. (1):

$$\text{Cross-Entropy Loss} = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (1)$$

where C is the total number of classes, y_i is the one-hot encoded ground-truth label for class i , and $\hat{y}_i$ is the model's predicted probability for that class.

To further enhance classification accuracy with minimal computational cost, the model integrates the SimAM attention module [9]. This module computes attention weights for features without requiring extra parameters. The process is defined by Eqs. (2)–(5):

$$x_{\text{minus_mu_square}} = \left(x - \frac{1}{hw} \sum_{i=1}^h \sum_{j=1}^w x_{ij} \right)^2 \quad (2)$$

$$\sigma^2 = \frac{1}{hw - 1} \sum_{i=1}^h \sum_{j=1}^w x_{\text{minus_mu_square}_{ij}} + \lambda \quad (3)$$

$$y = \frac{x_{\text{minus_mu_square}}}{4\sigma^2} + 0.5 \quad (4)$$

$$\text{output} = x \cdot \sigma(y) \quad (5)$$

here, x represents the input feature map with height h and width w . The term λ is a regularization parameter. The final output is derived from an element-wise multiplication ($\cdot$) of the input x with attention weights, which are obtained by passing the energy function result y through a sigmoid function $\sigma(\cdot)$.

The final severity classification model utilizes a hybrid loss function that combines Cross-Entropy Loss and Focal Loss to balance learning between common and rare severity classes. The final loss is a weighted sum, defined as in Eq. (6):

$$\text{HybridLoss} = w_{CE} \cdot L_{CE} + w_{Focal} \cdot L_{Focal} \quad (6)$$

where L_{CE} and L_{Focal} are the Cross-Entropy and Focal Loss values, respectively, and w_{CE} and w_{Focal} are the corresponding weights used to balance their contributions.

3.7 Embedded System Deployment

For deployment, the system utilizes the key hardware components detailed in Fig. 6. It runs on the NVIDIA Jetson AGX Orin embedded platform, accelerated using NVIDIA TensorRT [23] to achieve real-time inference speeds. Geographic localization of defects is enabled by an integrated GPS module, connected to the platform's GPIO pins. Real-time detection results, including precise locations, are systematically logged and analyzed, then visualized on an interactive web interface to improve usability and support efficient decision-making by road maintenance personnel.

4 Dataset and Preprocessing

4.1 Comprehensive Multi-Defect Dataset

The primary dataset for this study is the Road Damage Dataset 2022 (RDD2022) [5], containing 26,336 images collected across Japan, India, the Czech Republic, and the United States. Key features include

annotations for four primary defect types (longitudinal, transverse, alligator cracks, and potholes; see Fig. 7 for examples) crucial for safety analysis, as well as significant geographical and environmental diversity. This diversity is essential for developing robust models that generalize well to real-world conditions. The dataset's systematic labeling captures defect form and severity variations across different lighting and surface types.



Figure 6: Deployment platform and peripheral hardware.

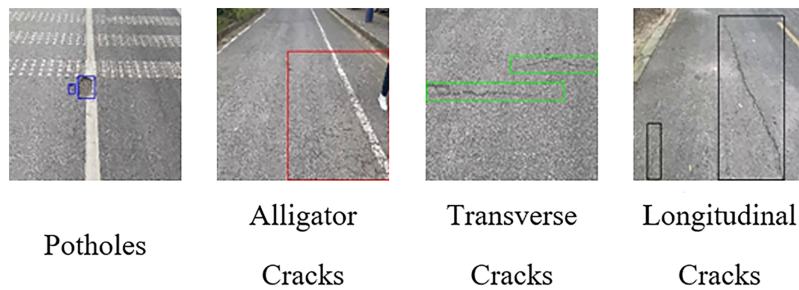


Figure 7: Road damage dataset RDD2022 [5].

Moreover, additional datasets featuring structural elements were integrated alongside the primary defect data to ensure comprehensive coverage relevant to practical road maintenance. These focused on common yet critical components like expansion joints, speed bumps, and manholes (see Fig. 8 for examples), which often require monitoring. Specifically, 844 speed bump images, 964 manhole images, and 565 bridge expansion joint images were added. This targeted integration provides data on elements impacting vehicle safety and operational stability.

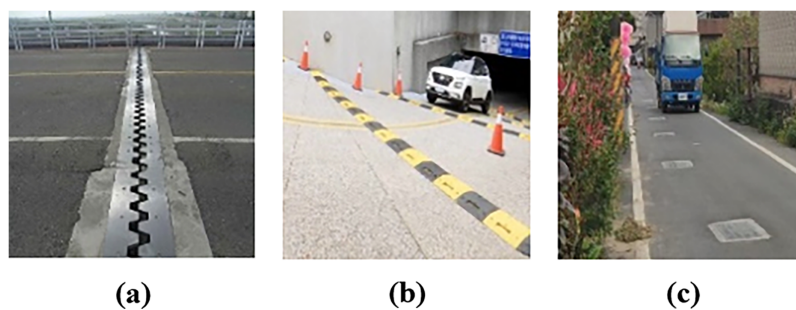


Figure 8: Additional dataset integration. (a) Expansion joints. (b) Speed bumps. (c) Manholes.

4.2 Construction of Severity Classification Dataset

Recognizing that RDD2022 lacks severity labels, we created a dedicated dataset for this classification task. The process involved cropping identified defects from RDD2022 images, standardizing them, and manually assigning severity labels (Minor, Moderate, Severe; see Fig. 9 for illustrative examples) based on

highway maintenance guidelines [7]. To ensure structural validity, severity categories (Minor, Moderate, Severe) are strictly anchored to the engineering frameworks of the ASTM D6433 pavement condition index criteria and the Federal Highway Administration (FHWA) distress identification guidance manual. Localized image variations, such as patch-level grayscale standard deviations, were utilized strictly as a secondary numerical sorting aid to verify feature uniformity. The resulting dataset comprises 4270 labeled severity patches—1563 Minor, 1494 Moderate, and 1213 Severe—partitioned into 3802 training and 468 validation samples.



Figure 9: Three levels of road damage severity.

Annotation Protocol & Quality Control: The dataset development utilized six trained domain annotators operating under the structural guidelines of the VGG Image Annotator (VIA) for binary mask definitions and standard bounding box labeling. The pipeline implemented a strict multi-pass quality-control hierarchy where two senior research specialists verified or re-annotated disputed samples. The severity gradings of the six annotators were consolidated into the final three-level consensus labels; because the original per-annotator records were not retained, a formal inter-annotator agreement statistic such as Fleiss' Kappa is not reported, and label reliability instead rests on this consensus consolidation together with the multi-pass review described above.

Road Segmentation Dataset: A specialized segmentation dataset was meticulously compiled to improve detection accuracy through focused feature isolation. Sourced mainly from real-world road imagery across Taichung City and supplemented with representative images from RDD2022, the dataset aims for broad applicability. Annotations were created using the VGG Image Annotator (VIA) to produce accurate binary masks distinguishing drivable surfaces from non-drivable elements (sidewalks, obstacles, vehicles, etc.) [24]. Emphasis was placed on including challenging conditions like shadows, diverse lighting, and obstructions. Fig. 10 provides a visual explanation of this process, showing how a binary mask isolates the drivable surface to improve subsequent defect detection.

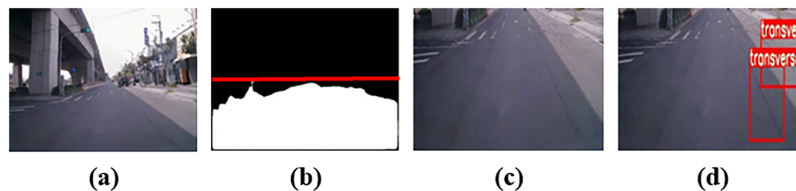


Figure 10: U-Net explanation for road segmentation is shown: (a) Input image, (b) Road segmentation, (c) Cropping above the uppermost road-region boundary, (d) Recognition after segmentation.

4.3 Data Preprocessing and Augmentation

All datasets underwent extensive preprocessing to optimize model performance and ensure consistent input quality. Initial steps involved resizing images to standardized dimensions— 640×640 pixels for detection and segmentation, and 224×224 for classification—balancing computational efficiency with accuracy. Subsequent preprocessing included histogram equalization for contrast enhancement and intensity normalization for consistency across diverse image sources.

5 Results

5.1 Hardware and Software Environment with Parameter Settings

All models were implemented using PyTorch 1.13.1 and trained on a high-performance workstation equipped with an NVIDIA GeForce RTX 4090 GPU and an Intel i5-14600K CPU, running on CUDA version 11.8. For model training, the Adam optimizer was consistently utilized across all models. The entire framework was subsequently deployed on the NVIDIA Jetson AGX Orin embedded platform to verify its real-time performance.

Key hyperparameters were individually optimized for each model to ensure the best performance.

- For the FAST-UNet segmentation model, we used a batch size of 16, a learning rate of 1×10^{-3} , and trained for 200 epochs with an input image resolution of 640×640 pixels.
- The YOLOv11_SDIDC detection model was trained for 300 epochs with a batch size of 16, a learning rate of 1×10^{-4} , and an input size of 640×640 pixels.
- The Improved-MobileNetV3 classification model was also trained for 300 epochs but with a batch size of 32, a learning rate of 1×10^{-4} , and an input resolution of 224×224 pixels.

Finally, all models were exported to the ONNX format and accelerated using NVIDIA TensorRT for efficient deployment.

5.2 Evaluation Metrics

The performance of the proposed framework was evaluated using a task-specific set of metrics covering road defect detection, road damage severity classification, semantic segmentation, and computational efficiency. The selected metrics were used to assess both predictive performance and the computational suitability of the proposed models for embedded deployment.

For road defect detection, Recall, Precision, and mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP50) were used to evaluate detection performance. Recall measures the proportion of ground-truth defect instances that are correctly detected, while Precision measures the proportion of predicted defect instances that are correctly identified. Their definitions are given in Eqs. (7) and (8), respectively:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

where TP , FP , and FN denote the numbers of true positives, false positives, and false negatives, respectively. For road damage severity classification, Recall, Precision, and F1-Score were used to evaluate the classification performance. While Recall and Precision measure the ability of the model to correctly identify severity classes and avoid incorrect class assignments, respectively, the F1-Score provides a harmonic mean of Precision and

Recall. The F1-Score is defined in Eq. (9). The use of F1-Score provides a single summary measure of the balance between Precision and Recall for severity classification. For object detection, Average Precision (AP) is computed from the precision-recall relationship as defined in Eq. (10), and mAP50 is obtained by averaging the AP values across all defect classes at an IoU threshold of 0.5, as defined in Eq. (11). Therefore, mAP50 provides an aggregate measure of detection performance across the evaluated defect categories.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

$$AP = \int_0^1 Precision(r) dr \quad (10)$$

$$mAP50 = \frac{1}{C} \sum_{c=1}^C AP_c \quad (11)$$

For road semantic segmentation, Recall, Precision, and the Dice Coefficient were used to evaluate pixel-level segmentation performance. Recall and Precision characterize the ability of the segmentation model to correctly identify road pixels and minimize incorrect road-pixel predictions, respectively. The Dice Coefficient measures the overlap between the predicted pixel set A and the ground-truth pixel set B , as defined in Eq. (12). A higher Dice value indicates greater agreement between the predicted road region and the corresponding ground-truth region.

$$Dice\ Coefficient = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (12)$$

For model efficiency and real-time deployment, Parameter Count, GFLOPs, and Frames Per Second (FPS) were considered. Parameter Count represents the total number of learnable parameters and is used to indicate model size, whereas GFLOPs represents the approximate computational complexity of a model. Lower Parameter Count and GFLOPs indicate a smaller computational footprint. FPS measures the inference throughput of the system and is calculated according to Eq. (13), where *Total Frames* represents the number of processed frames and *Processing Time* denotes the corresponding processing time in seconds. Higher FPS indicates greater inference throughput and better suitability for real-time applications.

$$FPS = \frac{Total\ Frames}{Processing\ Time\ (seconds)} \quad (13)$$

Overall, the metrics were interpreted according to the objective of each evaluation task. Higher Recall, Precision, F1-Score, mAP50, and Dice values indicate better predictive performance, whereas lower Parameter Count and GFLOPs indicate greater computational efficiency. Higher FPS indicates faster inference throughput. Together, these metrics provide complementary information on predictive performance and computational efficiency, allowing the proposed framework to be evaluated from both accuracy and embedded-deployment perspectives.

5.3 Model Comparison

Three segmentation architectures—U-Net, Lite-UNet, and the proposed FAST-UNet—were quantitatively and qualitatively compared. The numerical results, presented in Table 2, evaluate the models based on Recall, Precision, and the Dice coefficient. As shown in the table, FAST-UNet outperformed the other models, achieving the highest Dice score (0.986) while having the fewest parameters (0.31M), making it optimal

for deployment in embedded systems. Fig. 11 provides a qualitative comparison, visually demonstrating FAST-UNet's superior segmentation accuracy on challenging road scenes.

Table 2: Road segmentation experiment results. R: Recall, P: Precision, D: Dice, #P(M): Number of parameters in millions.

Model	R	P	D	#P(M)
UNet [17]	0.927	0.925	0.926	31.0
Lite-UNet [15]	0.970	0.966	0.968	1.95
Proposed FAST-UNet	0.987	0.984	0.986	0.31

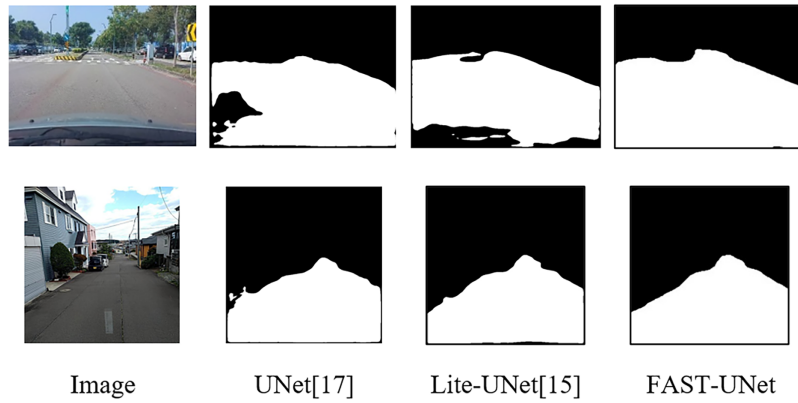


Figure 11: Segmentation experiment results of road segmentation models on specific road images.

The performance gain is attributed to its encoder-decoder structure enhanced by inverted residual blocks and SE (Squeeze-and-Excitation) modules [20], enabling effective spatial feature reuse and lightweight processing.

5.3.1 Object Detection Model Comparison

To evaluate its performance, YOLOv11_SDIDC was benchmarked against several state-of-the-art models, including YOLOv7 [4], YOLOv8n [25], YOLOv9-T [26], YOLOv10n [27], and YOLOv11n [16]. All models were trained and evaluated using the same RDD2022 dataset under the same evaluation protocol. The comprehensive comparison results are presented in Table 3. The results show that the proposed YOLOv11_SDIDC does not achieve the highest detection performance across all accuracy-related metrics. Its Recall of 0.674 is lower than that of YOLOv7 (0.728), while its Precision of 0.750 is lower than that of YOLOv11n (0.775). Similarly, its mAP50 of 0.722 is lower than the best-performing YOLOv7 (0.760), although it is comparable to YOLOv8n, which also achieves an mAP50 of 0.722.

Despite not achieving the highest accuracy, YOLOv11_SDIDC provides a substantial advantage in computational efficiency. It has the lowest parameter count (1.99M) and the lowest computational complexity (5.4 GFLOPs) among all compared models. Compared with YOLOv11n, for example, the proposed model reduces the parameter count from 2.59M to 1.99M and GFLOPs from 6.4 to 5.4, while maintaining a comparable mAP50 (0.722 vs. 0.720). These results indicate that the main advantage of YOLOv11_SDIDC is not universal superiority in detection accuracy, but a favorable trade-off between detection performance and computational efficiency.

Table 3: YOLO model comparison results.

Model	Recall	Precision	mAP50	Parameters	GFLOPs
YOLOv7 [4]	0.728	0.740	0.760	37,212,738	105
YOLOv8n [25]	0.681	0.756	0.722	3,011,628	8.19
YOLOv9-T [26]	0.658	0.757	0.646	2,661,626	11.0
YOLOv10n [27]	0.665	0.728	0.711	2,708,600	8.4
YOLOv11n [16]	0.663	0.775	0.720	2,591,205	6.4
Proposed YOLOv11_SDIDC	0.674	0.750	0.722	1,993,305	5.4

This trade-off is particularly relevant for embedded and real-time road-damage monitoring applications, where computational resources are constrained. The architectural modifications, including the SCDown downsampling module and InceptionNext-based feature extraction [21], are intended to improve feature representation while maintaining a compact model architecture. Therefore, the results demonstrate that YOLOv11_SDIDC should be viewed as a lightweight alternative that maintains competitive detection performance while substantially reducing the computational footprint rather than as a model that consistently outperforms all existing architectures.

5.3.2 Classification Model Comparison

The performance of the proposed Enhanced MobileNetV3 with SimAM was benchmarked against several established classification models, with the results presented in Table 4. Similar to the object detection results, the proposed model does not achieve the highest classification performance across all compared architectures. VGG19 achieves the highest F1-Score of 0.871, which is higher than the 0.811 obtained by the proposed Enhanced MobileNetV3. This result indicates that the proposed model does not outperform the larger VGG19 architecture in terms of classification accuracy.

Table 4: Road damage severity model training and validation results. R: Recall, P: Precision, F1: F1-Score, #P(M): Number of parameters in millions.

Model	R	P	F1	#P(M)
MobileNetv2 [19]	0.781	0.802	0.791	0.688
MobileNetv3_Small [8]	0.788	0.782	0.786	0.570
MobileNetv3_large [8]	0.792	0.789	0.787	1.41
AlexNet [28]	0.806	0.783	0.801	58.2
VGG16 [29]	0.787	0.774	0.779	134
VGG19 [29]	0.875	0.869	0.871	139
Enhanced MobileNetv3	0.812	0.809	0.811	1.02

However, the proposed model achieves competitive performance with a substantially smaller parameter count. With an F1-Score of 0.811, the Enhanced MobileNetV3 outperforms MobileNetv2 (0.791), MobileNetv3 Small (0.786), MobileNetv3 Large (0.787), AlexNet (0.801), and VGG16 (0.779), while requiring only 1.02M parameters. In comparison, VGG19 requires approximately 139M parameters to achieve an F1-Score of 0.871. Thus, the proposed model reduces the parameter count by approximately 136-fold relative to VGG19, while retaining a reasonably competitive classification performance.

Compared with the baseline MobileNetv3 Large model, the proposed Enhanced MobileNetV3 also improves the F1-Score from 0.787 to 0.811 while reducing the parameter count from 1.41M to 1.02M. This improvement is attributed to the integration of the parameter-free SimAM attention module, which enhances the representation of informative features for severity assessment without introducing a substantial increase in model size.

Overall, the results in Table 4 indicate that the proposed model should not be considered superior to all compared architectures in terms of classification accuracy. Instead, its main advantage lies in achieving competitive classification performance with substantially fewer parameters, making it a more suitable candidate for resource-constrained deployment.

5.3.3 Tracking Results

The integration of ByteTrack provided more reliable cross-frame association of static defects and substantially reduced duplicate detections (quantified in Section 5.5). Its two-stage matching strategy, which considers both high- and low-confidence detections, allowed more stable object identity tracking even under occlusion and motion blur. This makes it highly suitable for real-time road defect monitoring in dynamic environments.

5.4 Ablation Study

5.4.1 Classification Ablation Study

An ablation study was conducted on our proposed MobileNetV3 model to examine the individual and combined contributions of the SimAM attention mechanism [9] and a hybrid loss function (Focal Loss [10] + Cross-Entropy [11]). The experiments, with detailed results presented in Table 5, included four configurations:

- Baseline (MobileNetV3): Without SimAM or hybrid loss.
- SimAM only: Improved spatial focus, aiding in ambiguous region discrimination.
- Hybrid loss only: Addressed class imbalance in severity-level classification.
- Combined setup: Delivered the highest classification accuracy and fastest inference speed across test resolutions.

Table 5: Ablation research training and validation results.

Model	Recall	Precision	F1-Score	Parameters	GFLOPs
MobileNetv3_large [8]	0.792	0.789	0.787	1,412,363	2.49
MobileNetv3_large_SimAM	0.788	0.805	0.786	1,021,915	2.47
MobileNetv3_large_HybridLoss	0.761	0.788	0.755	1,412,363	2.49
Proposed MobileNetv3_large_SimAM_HybridLoss	0.812	0.809	0.811	1,021,915	2.47

Individually, neither SimAM (F1 = 0.786) nor the hybrid loss (F1 = 0.755) improves upon the baseline (F1 = 0.787). However, their combination raises the F1-Score to 0.811, indicating a complementary (synergistic) effect: SimAM sharpens feature-level discrimination of ambiguous regions, while the hybrid loss rebalances the optimization across severity classes; only when applied jointly do these two mechanisms yield a net gain.

5.4.2 Detection and Segmentation Architecture Ablation Studies

To rigorously verify the structural efficacy of the customized modifications introduced in our cascaded framework, comprehensive component-level ablation experiments were conducted independently for both the YOLOv11_SDIDC detector and the FAST-UNet segmenter.

First, the architectural components of the YOLOv11_SDIDC detector were isolated to evaluate the individual contributions of the SCDown modules and the C3k2_IDC/InceptionNext blocks. The quantitative outcomes of this analysis are detailed in [Table 6](#).

Table 6: YOLOv11_SDIDC detector structural component ablation study.

Configuration Variants	Module Additions	Recall	Precision	mAP50	Parameters	GFLOPs
Baseline Detector	YOLOv11n Baseline	0.663	0.775	0.720	2.59M	6.4
Structural Var. A	+ SCDown Modules Only	0.665	0.751	0.706	2.55M	5.8
Proposed Optimal	+ SCDown + C3k2_IDC Blocks	0.674	0.750	0.722	1.99M	5.4

As demonstrated in [Table 6](#), replacing the standard convolutional downsampling layers with the customized SCDown modules (Structural Variant A) yields a significant reduction in computational complexity, successfully contracting both the parameter footprint and total GFLOPs. Furthermore, the integration of the C3k2_IDC blocks in our proposed optimal configuration effectively compensates for potential feature representation losses during aggressive edge downsampling. By leveraging multi-scale receptive fields from the integrated InceptionNeXt design, the final detector achieves an optimal trade-off: it elevates the mAP50 to 0.722 while simultaneously minimizing the overall detector parameters down to a highly efficient 1.99M, making it exceptionally lightweight for real-time edge processing.

Second, a parallel ablation study was executed on the FAST-UNet segmentation model to isolate the operational benefits of embedding MobileNetV2 inverted residuals and Squeeze-and-Excitation (SE) attention mechanisms into a classic U-Net topology. The empirical performance shifts are summarized in [Table 7](#).

Table 7: FAST-UNet segmenter structural component ablation study.

Model Variants	Structural Segments	Recall	Precision	Dice Coefficient	Parameters
Seg Baseline	Standard U-Net Architecture	0.927	0.925	0.926	31.0M
Structural Var. B	+ MobileNetV2 Inverted Residuals	0.951	0.931	0.941	0.31M
Proposed Optimal	+ Inverted Residuals + SE Attention	0.987	0.984	0.986	0.31M

The empirical results in [Table 7](#) reveal an extraordinary structural optimization in the segmentation pipeline. The standard baseline U-Net, while visually effective, contains an un-deployable 31.0M parameters, which induces substantial execution latency on edge hardware. Substituting the standard heavy convolutional blocks with MobileNetV2 inverted linear bottlenecks (Structural Variant B) drastically reduces the network parameter size by over 98%. Crucially, the addition of the SE attention layer in the proposed optimal configuration dynamically recalibrates channel-wise feature dependencies. This architectural combination allows FAST-UNet to achieve a superior Dice Coefficient of 0.986 and an overall parameter footprint of just 0.31M. This proves that high-fidelity road region extraction can be achieved with minimal resource expenditure on constrained transport computing nodes.

5.5 Comprehensive Systems Performance and End-to-End Edge Profiling

To validate the operational performance of the complete road safety inspection framework, the integrated system was benchmarked on the NVIDIA Jetson AGX Orin platform under realistic on-vehicle deployment configurations. This evaluation measures the performance effects of real-world vehicular vibrations, driving speeds, and continuous run profiles.

End-to-End Pipeline Latency and Efficiency: When deployed using optimized TensorRT engines running in FP16 precision with a 4 MB allocation configuration, the total pipeline latency is 25.81 ms, supporting an operational throughput of 38.74 FPS. Measured on the same TensorRT engine, the constituent per-module latencies were 20.06 ms for the YOLOv11_SDIDC detector, 5.69 ms for the MobileNetV3 severity classifier, and 0.07 ms for the ByteTrack association step, which sum to the reported end-to-end latency. This throughput is highly sufficient to handle real-time imaging streams without dropping sequential frames at normal driving speeds. Peak hardware performance logs acquired via system utilities indicate a stable power draw of 7.2 W (peak on the SoC compute rails) and 14.41 W aggregate measured across all rails via tegrastats, and memory allocations bounded within 1.63 GB of edge memory. Thermal behavior was profiled over a sustained 200-frame MAXN profiling run, during which the GPU held its maximum 2.2 GHz clock with no computational throttling observed; a longer multi-hour endurance soak is planned as further validation.

Quantitative Defect Tracking and Deduplication: To quantify the benefit of tracking during vehicle movement: because each static defect is observed across many frames by the moving camera, non-tracking detection registers multiple duplicate counts of a single defect. Incorporating ByteTrack, which adds only 0.07 ms per frame, lowered the duplicate-detection rate of identical static features from an un-tracked rate of 88.89% down to an optimal tracked duplicate rate of 25.0%, providing the basis for GPS-based deduplication during coordinate logging.

Pipeline Error Propagation Metrics: To measure down-pipeline dependency errors, severity classification accuracy was isolated under conditioned pipeline states. When defects were perfectly framed by the YOLOv11_SDIDC detection layer, the severity network achieved a top-tier classification accuracy of 90.38%. Conversely, when input geometries were partially cut off by partial detector overlaps, classification accuracies shifted down to 49.15%, demonstrating the importance of bounding box precision on down-stream classification performance.

6 Conclusions

This study introduces an integrated and lightweight deep learning framework for real-time road defect detection, classification, segmentation, and tracking, optimized for deployment on embedded systems. The proposed pipeline leverages FAST-UNet for road segmentation, YOLOv11_SDIDC for multi-class defect detection [16], an enhanced MobileNetV3 with SimAM for severity classification [8,9], and ByteTrack for robust temporal tracking [12]. Deployed on the NVIDIA Jetson AGX Orin platform [18] and accelerated with TensorRT [23], the system delivers low-latency inference and competitive predictive performance, demonstrating its practicality for real-time smart city applications.

Experimental results demonstrate that the proposed models provide competitive predictive performance while achieving substantial computational efficiency. In particular, YOLOv11_SDIDC achieves the lowest parameter count (1.99M) and GFLOPs (5.4) among the compared YOLO models while maintaining a competitive mAP50 of 0.722. Although some existing models achieve higher individual detection or classification metrics, the proposed models provide a favorable balance between predictive performance and computational efficiency. Their modular architecture supports independent optimization of each component, offering flexibility for future system upgrades.

While our unified framework demonstrates robust edge-computing performance, its operational accuracy depends on specific environmental and physical constraints. First, the cascading deep learning layers are highly sensitive to camera mounting angles and vehicle height variations; non-standard perspectives can cause geometric distortions that affect segmentation contours and bounding boxes. Second, our on-vehicle validation covered a 0–50 km/h accelerate-cruise-decelerate profile; driving speeds beyond this tested range, combined with rough road conditions, can introduce localized motion blur and lens vibrations, leading to transient tracking losses and increased ID switching. Third, although the system generalizes well across typical urban asphalt roads, its cross-country transferability and accuracy on unstructured gravel pathways or highly customized concrete expansion styles remain unverified. Moreover, the present evaluation is scoped to standard daytime illumination; robust operation under low-light, night-time, and adverse-weather conditions was not characterized in this study and is therefore designated as a primary direction for future work. Finally, the specialized multi-task structural extensions and our 4270-sample distress severity dataset are subject to institutional data-sharing constraints and are currently provided strictly upon reasonable academic request, which may limit open cross-validation. Future development cycles will focus on integrating local structural enhancement networks and domain-adaptive tuning layers to systematically overcome these boundary challenges.

The system's design is well-suited for adaptation to broader infrastructure applications, including bridge deck inspections, airport runway monitoring, and urban utility mapping. Integration with additional sensing modalities, such as LiDAR or thermal imaging, is expected to further enhance detection precision and contextual awareness.

Overall, this work presents a comprehensive, efficient, and scalable solution for automated infrastructure monitoring. The proposed system provides a practical approach to real-time road condition assessment and contributes to the development of intelligent transportation systems and smart urban environments [4,10].

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the National Science and Technology Council (NSTC), Taiwan (NSTC 114-2221-E-035-076- and 115-2420-H-006-001-).

Author Contributions: The first draft was written by Feng-Cheng Lin and Jia-Yan Lin. The authors confirm contribution to the paper as follows: Conceptualization, methodology, writing—original draft preparation, Feng-Cheng Lin and Jia-Yan Lin; software, investigation, Jia-Yan Lin and Feng-Cheng Lin; data curation, validation, Feng-Cheng Lin, Chun-Yu Hung and Wijaya Wijaya; funding acquisition, supervision, Feng-Cheng Lin. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request. The source code for this study is available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

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

References

1. Silva LA, Leithardt VRQ, Batista VFL, Villarrubia González G, De Paz Santana JF. Automated road damage detection using UAV images and deep learning techniques. *IEEE Access*. 2023;11:62918–31. doi:10.1109/ACCESS.2023.3287770.

2. Bochkovskiy A, Wang CY, Liao HM. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934. 2020. doi:10.48550/arXiv.2004.10934.
3. Jocher G, Stoken A, Borovec J, Changyu L, Hogan A, Diaconu L, et al. YOLOv5. 2020 [cited 2026 Sep 1]. Available from: <https://github.com/ultralytics/yolov5>.
4. Wang CY, Bochkovskiy A, Liao HM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv:2207.02696. 2022. doi:10.48550/arXiv.2207.02696.
5. Arya D, Maeda H, Ghosh SK, Toshniwal D, Sekimoto Y. RDD2022: a multi-national image dataset for automatic road damage detection. Geosci Data J. 2024;11(4):846–62. doi:10.1002/gdj3.260.
6. Zou C, Huang P, Wang S, Wang C, Wang H. Pavement defect detection algorithm based on improved YOLOv7 complex background. IEEE Access. 2024;12(3):32870–80. doi:10.1109/ACCESS.2024.3370604.
7. Song W, Jia G, Jia D, Zhu H. Automatic pavement crack detection and classification using multiscale feature attention network. IEEE Access. 2019;7:171001–12. doi:10.1109/ACCESS.2019.2956191.
8. Howard A, Sandler M, Chen B, Wang W, Chen LC, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 1314–24. doi:10.1109/iccv.2019.00140.
9. Yang L, Zhang RY, Li L, Xie X. SimAM: a simple, parameter-free attention module for convolutional neural networks. In: Proceedings of the 38th International Conference on Machine Learning; 2021 Jul 18–24; Online. p. 11863–74.
10. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):318–27. doi:10.1109/TPAMI.2018.2858826.
11. Mao A, Mohri M, Zhong Y. Cross-entropy loss functions: theoretical analysis and applications. In: Proceedings of the 40th International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. p. 23803–28.
12. Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, et al. ByteTrack: multi-object tracking by associating every detection box. In: Proceedings of the Computer Vision—ECCV 2022. New York, NY, USA: The Association for Computing Machinery (ACM); 2022. p. 1–21. doi:10.1007/978-3-031-20047-2_1.
13. Lin C, Tian D, Duan X, Zhou J, Zhao D, Cao D. DA-RDD: toward domain adaptive road damage detection across different countries. IEEE Trans Intell Transp Syst. 2022;24(3):3091–103. doi:10.1109/TITS.2022.3221067.
14. 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. 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
15. El-Assiouti HS, El-Saadawy H, Al-Berry MN, Tolba MF. Lite-SRGAN and lite-UNet: toward fast and accurate image super-resolution, segmentation, and localization for plant leaf diseases. IEEE Access. 2023;11:67498–517. doi:10.1109/ACCESS.2023.3289750.
16. Khanam R, Hussain M. YOLOv11: an overview of the key architectural enhancements. arXiv:2410.17725. 2024. doi:10.48550/arXiv.2410.17725.
17. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015. Cham, Switzerland: Springer; 2015. p. 234–41. doi:10.1007/978-3-319-24574-4_28.
18. Nvidia Corporation. Jetson AGX Orin developer kit user guide. 2025 [cited 2026 Sep 1]. Available from: <https://docs.nvidia.com/jetson/agx-orin-devkit/user-guide/latest/index.html>.
19. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 4510–20. doi:10.1109/CVPR.2018.00474.
20. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 7132–41. doi:10.1109/CVPR.2018.00745.
21. Yu W, Zhou P, Yan S, Wang X. InceptionNeXt: when inception meets ConvNeXt. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 5672–83. doi:10.1109/CVPR52733.2024.00542.

22. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IoU loss: faster and better learning for bounding box regression. AAAI Conf Artif Intell. 2020;34(7):12993–3000. doi:10.1609/aaai.v34i07.6999.
23. Nvidia Corporation. TensorRT: high performance deep learning inference platform. 2025 [cited 2026 Sep 1]. Available from: <https://developer.nvidia.com/tensorrt>.
24. Dutta A, Zisserman A. The VGG image annotator (VIA). In: Proceedings of the 27th ACM International Conference on Multimedia. New York, NY, USA: The Association for Computing Machinery (ACM); 2019. p. 2276–9.
25. Ultralytics. YOLOv8. 2023 [cited 2026 Sep 1]. Available from: <https://docs.ultralytics.com/>.
26. Wang CY, Yeh IH, Mark Liao HY. YOLOv9: learning what you want to learn using programmable gradient information. In: Proceedings of the Computer Vision—ECCV 2024. Cham, Switzerland: Springer; 2025. p. 1–21. doi:10.1007/978-3-031-72751-1_1.
27. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. YOLOv10: real-time end-to-end object detection. arXiv:2405.14458. 2024. doi:10.48550/arXiv.2405.14458.
28. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90. doi:10.1145/3065386.
29. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. 2015. doi:10.48550/arXiv.1409.1556.