RESEARCH ARTICLE

Real-Time Detection of Road Anomalies for Integration in Rider Assistance Systems

Tiago Silva[1,*], **João Silva**[1,*], **António Sousa**[1,2,*] and **Vítor Filipe**[1,2,*]

[1] University of Trás-os-Montes and Alto Douro, 5001-801 Vila Real, Portugal
[2] INESC TEC - Institute for Systems and Computer Engineering, Technology and Science, 4200-465 Porto, Portugal

Abstract

Road safety has become an increasingly important concern and the integration of Advanced Rider Assistance Systems and Advanced Driver Assistance Systems plays a crucial role in preventing accidents. This work proposes a computer vision pipeline to automatically detect hazardous road anomalies—*loose gravel*, *potholes*, and *puddles*—from a motorcycle-mounted camera, targeting real-time operation on embedded edge devices. A hybrid dataset of 28764 annotated images was created by combining real-world photos, Blender-rendered synthetic scenes, and AI-generated images to improve diversity and coverage. Multiple state-of-the-art object detectors were trained and benchmarked, including the YOLOv5/7/11/12 families and the transformer-based RT-DETR architecture. While the RT-DETR model achieved the highest precision overall, its computational complexity and heavy resource requirements limited its suitability for real-time deployment on low-cost embedded platforms. Conversely, the YOLOv11n model demonstrated the best accuracy–efficiency trade-off, reaching mAP@0.5 = 0.872 at 320×320 with 0.045 s/frame on a Jetson Nano, while lighter variants remained viable on Raspberry Pi boards. Across classes, *gravel* was the most reliably detected, and operating points around a confidence threshold of $\tau \approx 0.31$ yielded balanced F1 scores up to 0.82. Although results show that automatic road-condition monitoring on affordable hardware is feasible, the prototype has not yet undergone on-road field trials. It does not include an integrated rider alert module or energy-use assessment. These gaps define the immediate roadmap for deployment.

Keywords: road monitoring, computer vision, object detection, edge devices, advanced rider assistance systems.

Academic Editor:
Javier Bajo

Submitted: 29 September 2025
Accepted: 19 November 2025
Published: 03 March 2026

Vol. 3, **No.** 1, 2026.
10.62762/TIS.2025.418469

***Corresponding authors:**
✉ Tiago Silva
tiagofsilva@utad.pt
✉ João Silva
andres@utad.pt
✉ António Sousa
amrs@utad.pt
✉ Vítor Filipe
vfilipe@utad.pt

1 Introduction

Road crashes remain a major public health problem worldwide. In the European Union, there were about 892000 recorded road accidents and 20634 fatalities in 2022 [1, 2]. These figures highlight the need for effective, scalable safety measures that can operate in

Citation
Silva, T., Silva, J., Sousa, A., & Filipe, V. (2026). Real-Time Detection of Road Anomalies for Integration in Rider Assistance Systems. *ICCK Transactions on Intelligent Systematics*, 3(1), 32–54.

real-world conditions.

Motorcyclists are especially vulnerable to surface-related hazards. Loose gravel, puddles, and potholes reduce tire traction and increase the risk of skidding and loss of control. According to Wu et al. [3], degraded road adhesion is a major cause of motorcycle loss-of-control (LOC) crashes, increasing the risk more than twentyfold when deterioration is encountered unexpectedly due to gravel, oil, or potholes. Although motorcycles represent only 1.5% of total road traffic, they account for nearly a quarter of all road crashes and have a 20–30 times higher risk of fatal or serious injury per kilometer compared to car drivers [3]. More recently, Lyu et al. [4] demonstrated that low pavement friction remains one of the most influential predictors of motorcycle crash frequency, with a 10-unit increase in friction (measured as SR40) reducing crashes by 19–36%. Conversely, segments with friction values below SR40=37 exhibit sharply higher accident rates, confirming that inadequate surface grip is a key factor in motorcycle instability. Furthermore, inconsistent friction across road segments and insufficient macrotexture significantly elevate crash risks, particularly on curves and intersections [4].

These findings underscore the critical role of road surface quality in motorcycle safety and highlight the need for automated systems capable of detecting surface anomalies in real time. Recent studies emphasize the value of comprehensive data and machine learning to identify risk patterns and support evidence-based interventions [5, 6]. In parallel, advances in computer vision and deep learning have enabled automated and increasingly reliable road-surface monitoring [7–10]. However, most existing approaches target car-based ADAS systems rather than motorcycle-oriented ARAS solutions, where stability and compactness impose stricter hardware and latency constraints. Additionally, current road-surface datasets often underrepresent critical hazards such as gravel or puddles, limiting model generalization to real-world environments. Few studies have evaluated the trade-off between detection accuracy and latency on low-cost embedded devices such as the `Raspberry Pi` or `Jetson Nano`, which are crucial for affordable deployment in two-wheeled vehicles.

This work presents a practical system to detect three critical road anomalies—*gravel*, *potholes*, and *puddles*—from a motorcycle-mounted camera, targeting real-time use on low-cost edge hardware. We build a diverse dataset that combines real-world images, Blender-generated synthetic scenes, and AI-synthesized imagery to improve coverage of rare and challenging cases. We then train and compare state-of-the-art object detectors (YOLO, SSD, R-CNN, and RT-DETR) and evaluate their accuracy–latency trade-offs on Raspberry Pi and Jetson Nano platforms.

This research addresses these limitations by proposing a unified, edge-oriented framework for real-time detection of critical road anomalies—*gravel*, *potholes*, and *puddles*—using camera data from motorcycle-mounted systems. The work introduces a hybrid dataset combining real, synthetic, and AI-generated imagery to enhance model robustness under diverse conditions and systematically benchmarks state-of-the-art detectors (YOLO and RT-DETR) on affordable embedded hardware.

The key **contributions and innovations** of this study are as follows:

- **Hybrid dataset creation:** a novel combination of real-world, Blender-simulated, and generative-AI images designed to address class imbalance and improve generalization across anomaly types.
- **Comprehensive model–hardware benchmarking:** first multi-device comparison of YOLO and transformer-based detectors on Raspberry Pi 4/5 and Jetson Nano, assessing both detection accuracy and latency.
- **Edge-optimized model analysis:** identification of optimal model–resolution trade-offs and inference configurations suitable for motorcycle-oriented ARAS systems.
- **Guidelines for real-time deployment:** practical insights on balancing accuracy, efficiency, and energy use to support future integration of AI-based rider assistance technologies.

The remainder of this article is organized as follows. Section 2 reviews related work in the field of road anomaly detection and rider assistance systems. Section 3 presents the methodology, including dataset development, model selection, and training procedures. Section 4 describes the hardware platforms considered, while Section 5 reports and analyzes the experimental results. Section 6 presents a detailed discussion of the findings, and Section 7 concludes the article by highlighting the main contributions and outlining future research directions.

2 Related Work

Road maintenance and the accurate detection of road surfaces are crucial to enhancing safety, optimizing infrastructure management, and facilitating the development of autonomous and assisted driving systems. The growing need to reduce accidents has accelerated the growth of Advanced Driver Assistance Systems (ADAS) and Advanced Rider Assistance Systems (ARAS), which integrate features such as adaptive cruise control, lane departure warnings, blind spot detection, anti-lock braking systems, and road condition monitoring [11]. While ADAS is mainly applied to cars, ARAS targets motorcycles and e-bikes, requiring less intrusive interventions to preserve stability. Both rely heavily on radar, cameras, and sensor fusion, reflecting a broader trend in Intelligent Transportation Systems (ITS) toward real-time, data-driven safety improvements [12].

Recent advances in road surface analysis highlight the importance of datasets and multimodal approaches. Large annotated datasets, combined with CNN-based models, have achieved high accuracy in classifying road conditions [7]. Meanwhile, audio-visual fusion strategies using VGG19 have demonstrated promising results for cost-effective maintenance [9]. Defect detection techniques have also advanced through lightweight neural networks such as LHA-Net [13], UAV-based applications with MS-YOLOv7 [14], and methods robust to lighting variability such as IlumiCrack [15]. Faster R-CNN and synthetic data augmentation have further improved performance under real-world constraints [16].

Applications have expanded from general road defect detection to more specific challenges, such as detecting gravel, puddles, and bumps. Transfer learning with ResNet and DenseNet has been effective for gravel classification [8]. In contrast, self-attention models such as AGSENet [17] and architectures like DCU-Net [18] have achieved high accuracy in water hazard segmentation. Hybrid approaches that combine deep learning with traditional machine learning have been proposed for pothole severity classification [19]. YOLO-based methods have also been adapted for resource-constrained environments, such as UAVs and embedded systems [20, 21]. Low-cost ADAS prototypes also integrate YOLO models with depth cameras for simultaneous detection of anomalies and traffic signs [11], while ARAS systems for e-bikes combine YOLO with gyroscopic and ultrasonic sensors to ensure rider safety [22].

Overall, the literature shows a rapid evolution from traditional computer vision toward deep learning, multimodal fusion, and edge-deployable solutions. These developments, reviewed extensively by Yu et al. [23], Dilek et al. [24], Safyari et al. [25], and Rathee et al. [26], collectively contribute to safer and more sustainable transportation systems by supporting proactive infrastructure maintenance, providing adaptive assistance to both drivers and riders, and delivering scalable solutions suitable for diverse environments and platforms.

Beyond establishing the safety motivation, prior work has begun to compare *accuracy–latency* trade-offs of road-defect detectors on embedded hardware, showing that lightweight one-stage models can achieve *near real-time* throughput with competitive accuracy on low-cost devices. For example, YOLO-based ADAS prototypes reach real-time inference on NVIDIA Jetson Nano while maintaining acceptable detection quality for bumps and traffic signs [11]; feasibility studies using smartphone video and Raspberry Pi report that compact CNN/YOLO variants offer practical latency on CPU-bound platforms, whereas heavier two-stage/transformer models remain constrained by inference time [27]. Edge-focused variants of YOLOv8 optimized for UAVs further illustrate how architectural tweaks can improve efficiency without sacrificing detection robustness under challenging conditions [20]. However, comprehensive evaluations targeted at motorcycle-centric anomalies (gravel, puddles, potholes) across multiple edge devices are still scarce. This gap motivates our systematic benchmark that jointly reports detection metrics and on-device latency on Raspberry Pi 4/5 and Jetson Nano, guiding operating points suitable for ARAS scenarios. Moreover, the current body of literature on motorcycle-oriented road anomaly detection remains limited, with only a few studies explicitly addressing the unique challenges of two-wheeled vehicles. As a result, available datasets and comparative analyses remain scarce, constraining the empirical foundation of existing research. This lack of prior work reinforces the relevance of the present study, as it contributes new experimental evidence and performance insights that can serve as a reference point for future developments in this emerging field.

3 Methodology

This section presents a structured and systematic approach to achieving the objectives outlined in the introduction. It details the methodologies used for data acquisition, including the collection and preparation of real, synthetic, and AI-generated images. The selection criteria and training protocols for state-of-the-art object detection models are described, along with the evaluation metrics used to assess their performance. The chapter also discusses optimization strategies for deploying these models on edge computing devices, specifically the Raspberry Pi and Jetson Nano. Finally, it outlines the implementation and real-world testing strategies employed to assess the practical application of the proposed system, with a focus on enhancing road safety for motorcyclists.

The methodological design directly operationalizes the study's main contributions outlined in Section 1. Specifically, it establishes (i) a hybrid data generation pipeline integrating real-world, simulated, and AI-synthesized imagery; (ii) a systematic benchmarking framework for evaluating object detection models under edge-device constraints; and (iii) a reproducible protocol for analysing detection performance versus computational efficiency. Together, these components ensure the experimental workflow is tightly aligned with the study's innovation goals.

3.1 Theoretical Foundations and Supporting Literature

The methodological framework adopted in this study is grounded in the fundamental theories of modern object detection, data generation, and model optimization for edge computing.

From an algorithmic perspective, contemporary object detectors can be categorized into two main paradigms: *two-stage* and *one-stage* architectures. Two-stage approaches, such as Faster R-CNN, first generate region proposals before classification and regression, achieving high accuracy but at greater computational cost [28]. In contrast, one-stage detectors such as SSD and the YOLO family predict bounding boxes and class probabilities in a single pass, enabling real-time performance [29]. Improvements in small-object detection have been made possible through Feature Pyramid Networks (FPN), which combine multi-scale feature representations to preserve both fine-grained spatial detail and semantic richness [30]. More recently, transformer-based architectures such as DETR and Deformable DETR have reformulated detection as a set prediction problem that leverages global self-attention mechanisms, simplifying post-processing and improving generalization [31, 32]. These theoretical advances justify the inclusion of YOLO and RT-DETR models in this work, as they embody efficient and accurate implementations of these principles.

Regarding dataset generation, this study follows established strategies in domain adaptation and synthetic data creation. The use of *domain randomization* [33] and *copy–paste augmentation* [34] enables better generalization to unseen real-world environments by randomizing lighting, textures, and object placement in synthetic scenes. Furthermore, generative modeling techniques such as Generative Adversarial Networks (GANs) [35], CycleGAN [36], StyleGAN [37], and latent diffusion models [38] enable the synthesis of high-quality, diverse images for training and validation. Simulation tools such as BlenderProc [39] and CARLA [40] are also widely used for photorealistic rendering with precise annotations, supporting the hybrid real–synthetic dataset strategy adopted in this research.

Finally, to ensure feasibility on low-cost embedded hardware, several *edge optimization* strategies were implemented. Structured channel pruning [41] and deep compression [42] reduce model complexity and memory requirements, while quantization-aware training (QAT) [43] enables integer-only inference (INT8) with minimal accuracy degradation. Knowledge distillation [44] further improves the generalization of lightweight models by transferring information from larger, high-capacity teacher networks to smaller student models. Together, these optimization methods minimize latency and energy consumption, enabling real-time inference on devices such as the Raspberry Pi and Jetson Nano.

In summary, the theoretical foundation of this study is supported by well-established frameworks in object detection, data generation, and edge optimization. These references reinforce the scientific rigor of the methodology and align the proposed system with state-of-the-art research in efficient deep learning for intelligent transportation systems.

3.2 Selection of Object Detection Models

To support the experimental evaluation of road anomaly detection in the context of motorcycle rider assistance, several state-of-the-art object detection models were selected and tested on edge computing devices. The selection process focused on models that offer a balance between detection accuracy and computational efficiency, making them suitable for deployment in real-time scenarios with constrained hardware, such as motorcycles equipped with embedded systems.

The chosen models span different architectural families, including one-stage detectors like YOLO and SSD, as well as more complex two-stage approaches such as R-CNN. In addition, transformer-based architectures, such as RT-DETR, were also included to assess the capabilities of newer attention-based mechanisms in edge deployments.

These models were selected based on several key criteria:

- **Real-time inference potential:** Since the goal is to assess detection performance in near real-time on platforms like Raspberry Pi and Jetson Nano, models with low latency and reduced computational load were prioritized.
- **Compatibility with edge deployment:** Lightweight versions of object detectors (e.g., YOLOv5n, YOLOv7-tiny, SSDLite) were chosen due to their suitability for ARM-based hardware and availability of pre-trained weights and export tools (e.g., ONNX, TensorRT).
- **Performance on small objects:** Road anomalies such as gravel and potholes often appear as small or partially occluded objects in the scene. YOLO and RT-DETR variants were included for their support of multiscale detection and recent enhancements in spatial sensitivity.
- **Architectural diversity:** By evaluating different detection paradigms — from CNN-based single-shot detectors (YOLO, SSD) to transformer-based architectures (RT-DETR) — the study aims to compare trade-offs in performance, inference time, and resource consumption across model families.

Inference time was selected as a primary evaluation metric, as it directly impacts the feasibility of deploying these models in embedded systems mounted on motorcycles. Additionally, accuracy, precision, recall, and mean average precision were used to quantify detection performance across three anomaly classes: gravel, potholes, and puddles.

This comprehensive benchmarking enables a deeper understanding of which detection models are most suitable for future integration in real-time rider assistance systems, even if a whole system is not implemented in this work.

3.2.1 SSDLite

SSDLite is a lightweight version of the Single Shot MultiBox Detector, specifically designed for mobile and embedded applications. It was introduced in the MobileNetV2 paper and modifies the original SSD architecture by replacing standard convolutions with depthwise separable convolutions. This change significantly reduces computational complexity while maintaining competitive accuracy. The model typically employs MobileNetV2 or MobileNetV3 as its backbone, benefiting from inverted residuals and linear bottlenecks that enhance efficiency without compromising the quality of feature extraction [45].

3.2.2 YOLO Family

YOLO (You Only Look Once) is a family of real-time, one-stage object detection models designed for speed and efficiency. Unlike traditional two-stage detectors, such as Faster R-CNN, which first generate region proposals before classifying objects, YOLO processes the entire image in a single forward pass, making it significantly faster. The YOLO architecture divides the image into a grid and predicts bounding boxes, class probabilities, and confidence scores for objects within each grid cell.

YOLOv3, introduced in 2018, improved upon previous versions by incorporating a stronger backbone, Darknet-53, which consists of 53 convolutional layers with residual connections. The model introduces multiscale detection, making predictions at three different scales to improve small-object detection. Instead of softmax activation, it uses independent logistic classifiers, enabling multi-label classification for objects within a single bounding box [46].

YOLOv5 was released by Ultralytics in 2020 and introduced a PyTorch-based implementation, making it more accessible for training and deployment. The model relies on CSPNet as its backbone, which reduces redundant gradient information while maintaining learning capability. The neck incorporates PANet to enhance feature fusion, and the detection head optimizes the bounding box prediction process [47].

YOLOv7 introduced E-ELAN for enhanced feature representation. Unlike previous versions, it incorporates model re-parameterization techniques to improve inference speed while maintaining high accuracy. The backbone maintains a highly optimized convolutional architecture, the neck integrates SPP to enhance multiscale object detection, and the detection head is designed for rapid inference [48].

YOLOv8, **YOLOv10**, and **YOLOv11** introduced additional architectural improvements, particularly in efficiency and scalability for real-time applications. YOLOv8 transitioned to an anchor-free detection mechanism, which reduces computation time and improves small-object detection. YOLOv10 focused on optimizing lightweight models for edge devices, while YOLOv11 introduced Transformer-based feature fusion to enhance context understanding. These models continue to push the boundaries of object detection for applications requiring high speed and accuracy [49–51].

YOLOv9 builds on the anchor-free approach of YOLOv8, introducing Dynamic Neural Architecture Search (DNAS) for adaptive scaling and Hybrid Efficient Attention Mechanisms (HEAM) to improve feature extraction. These enhancements boost accuracy while maintaining real-time efficiency, making YOLOv9 well-suited for fast, high-precision object detection [52].

YOLOv12 is the most recent version, at the time of development of this work, of the YOLO object detection framework, which focuses on attention mechanisms to improve real-time object detection. It introduces an attention-centric architecture that enhances feature extraction and prioritizes relevant regions, resulting in improved accuracy and efficiency in object detection, particularly in complex environments [53].

3.2.3 RT-DETR

RT-DETR (Real-Time Detection Transformer) is a state-of-the-art, transformer-based object detection architecture specifically designed to achieve a balance between detection accuracy and real-time inference speed. It builds upon the foundational DETR architecture by integrating optimizations aimed at reducing computational complexity while retaining the superior feature extraction and global attention capabilities of transformer-based models [54–56].

Unlike conventional convolutional models such as YOLO, which rely heavily on hand-crafted anchor boxes and local receptive fields, RT-DETR employs a fully end-to-end transformer design that enables global context modeling. A key innovation of RT-DETR is the use of dynamic anchor boxes, which adaptively adjust to the shape and scale of objects during training and inference. This flexibility allows the model to handle a wide range of object sizes and aspect ratios more effectively than static anchor-based methods.

To further enhance performance, RT-DETR introduces architectural refinements, including lightweight transformer encoders and decoders, anchor assignment strategies based on spatial and semantic consistency, and training optimizations such as the bag-of-freebies methodology. These improvements contribute to faster convergence during training and significant reductions in inference latency, making RT-DETR suitable for deployment on real-time systems and resource-constrained platforms.

RT-DETR's ability to unify the detection pipeline—eschewing the need for post-processing steps like non-maximum suppression (NMS)—also simplifies the detection framework. This end-to-end learning capability contributes to improved object localization and reduced false positives, particularly in crowded or complex scenes.

Overall, RT-DETR represents a significant advancement in object detection by combining the expressive power of transformers with the practical requirements of real-time performance. Its strong generalization capabilities and adaptability make it an appealing choice for applications such as autonomous driving, UAV monitoring, and mobile robotics.

3.2.4 Models Summary

Table 1 provides a comparative overview of several prominent object detection models utilized in computer vision. The table details each model's year of introduction, core architecture, backbone network, key distinguishing features, and corresponding references. This comparison facilitates an understanding of the evolution and characteristics of different object detection approaches, ranging from traditional convolutional networks to more recent transformer-based architectures. It highlights their specific design considerations and advancements.

3.3 Datasets

The datasets used in this study are organized according to a clear hierarchical structure designed to support both model benchmarking and task-specific anomaly detection. Two complementary datasets were employed, each serving a distinct purpose within the research framework.

First, the **BDD100K-Based Dataset** was used primarily as a *reference benchmark* to evaluate model inference performance under a wide range of driving environments, encompassing different lighting conditions, weather scenarios, and traffic densities. This dataset provides a robust foundation for assessing

Table 1. Comparison of object detection models.

Model	Year	Architecture	Backbone	Key Features	Ref.
SSDLite	2018	One-stage detector with depthwise separable convolutions	Mobile-NetV2, Mobile-NetV3	Optimized for mobile devices, reduced computational complexity	[45]
YOLOv3	2018	One-stage detector with Darknet-53 and multi-scale predictions	Darknet-53	Three-scale predictions, multi-label classification	[46]
YOLOv5	2020	One-stage PyTorch-based detector with CSPNet backbone	CSPNet	Efficient training and inference with PyTorch	[47]
YOLOv7	2022	One-stage detector with E-ELAN and SPP	E-ELAN	Model re-parameterization for speed improvement	[48]
YOLOv8	2023	One-stage anchor-free detector with CSPDarknet	CSPDarknet	Anchor-free detection, supports segmentation and pose estimation	[49]
YOLOv9	2024	One-stage anchor-free detector with improved CSPDarknet	Improved CSPDarknet	Anchor-free detection, integrates DNAS and HEAM	[52]
YOLOv10	2024	One-stage-detector with Optimized CSPNet, BiFPN, EfficientIoU Loss	Optimized CSPNet	Optimized for edge devices, quantization support	[50]
YOLOv11	2024	Transformer-based feature fusion, Sparse Convolutions	EfficientNet-inspired CSPNet	Self-attention mechanisms, sparse convolutions	[51]
YOLOv12	2025	Hybrid transformer-convolutional model with dynamic sparsity	HybridNet	Enhanced self-attention, adaptive feature selection for real-time inference	[53]
RT-DETRv2	2023	Transformer-based real-time detector with Vision Transformer backbone	Vision Transformer	Set-based detection, self-attention for context learning	[54]

general model performance and transferability to real-world driving contexts.

Second, the **Custom Hybrid Dataset** was developed specifically for the *road anomaly detection* task. This dataset is hierarchically organized into three anomaly categories—**gravel**, **potholes**, and **puddles**—each designed to capture distinct visual and structural surface irregularities. Among these, only the **gravel** class integrates images from three complementary sources: *real-world imagery*, collected from actual road environments; *synthetic imagery*, generated using Blender-based 3D simulations; and *AI-generated imagery*, produced with generative models to enhance data diversity and realism. In contrast, the **pothole** and **puddle** classes consist exclusively of *real-world images*, carefully selected to represent a wide range of visual conditions and environmental contexts.

This hierarchical organization ensures a coherent distinction between the benchmarking dataset (BDD100K-based) and the task-oriented dataset (Custom Hybrid), while also highlighting the specific composition and source diversity associated with each anomaly category.

3.3.1 BDD100K-Based Dataset

To thoroughly assess inference performance on the Raspberry Pi in diverse real-world scenarios relevant to autonomous driving, we created a specialized dataset using the comprehensive BDD100K database[1]. Autonomous driving requires robust perception across diverse geographical locations (urban, suburban, and rural) and challenging environmental conditions (rain, fog, snow, day and night, and varying traffic densities). The BDD100K dataset, which comprises 100,000 annotated videos and images with detailed object detection labels [57], provides an excellent basis for this evaluation.

We developed a custom evaluation dataset based on the publicly available BDD100K dataset to enable benchmarking of object detection models under a wide range of real-world driving conditions. Although BDD100K provides a vast collection of annotated images, it does not offer a structured subset focused on edge-case environmental conditions. To address this, we designed and implemented a custom script that programmatically filtered and organized the original dataset into targeted groups.

The script (developed in Python) parses the metadata associated with each image — specifically the `weather`, `timeofday`, and `scene` fields — to classify the data into 16 distinct combinations of environmental conditions. These combinations cover common and challenging scenarios, such as driving in clear or rainy weather, during the day or night, across highways or residential areas. Each resulting group includes up to 1,000 representative images extracted from the BDD100K training set. The selected images are physically copied into separate folders, and the corresponding metadata annotations are saved as `labels.json` files within each folder. This structure makes the dataset modular, interpretable, and practical for batch testing of model inference on resource-constrained platforms such as Raspberry Pi or Jetson Nano. This custom dataset, therefore, is not a new dataset in itself but a curated and categorized version of the BDD100K training set restructured specifically for model evaluation in diverse and controlled real-world scenarios. Figure 1 illustrates the organization process implemented.

The selection of these 16 unique combinations was strategically made to cover a range of common and challenging conditions for autonomous driving systems. Each category, containing approximately 1,000 images, provides a focused subset for evaluating the robustness and performance of inference models.

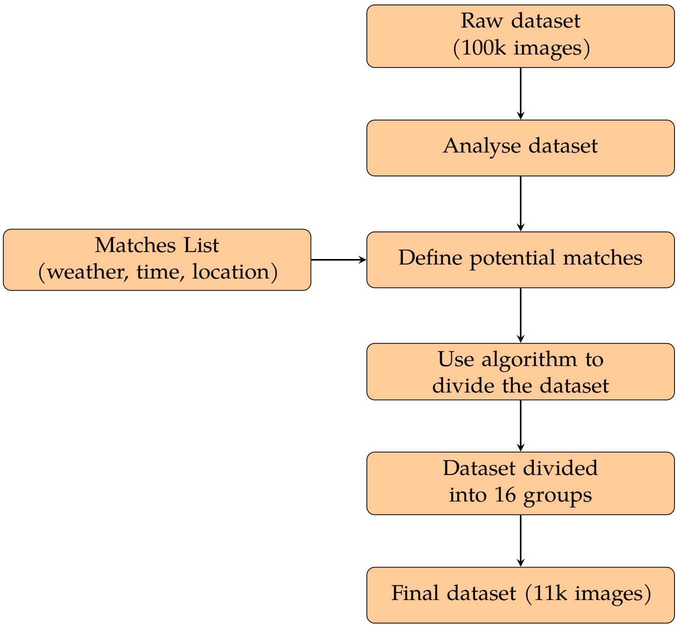

Figure 1. Dataset organization process.

3.3.2 *Dataset for Road Anomaly Detection*

The development of a high-quality dataset is a cornerstone of any successful machine learning project, particularly in the field of computer vision. In the context of road anomaly detection, the dataset must capture a wide range of road surface irregularities under diverse environmental and visual conditions. This is essential to ensure that models trained on such data can generalize effectively to real-world scenarios and maintain high accuracy in safety-critical applications such as autonomous driving, infrastructure monitoring, and smart mobility systems.

To construct a robust and representative dataset for this research, a hybrid image acquisition strategy was adopted, integrating three complementary methodologies: (1) real-world image acquisition from public sources, (2) synthetic image generation using the 3D modeling platform Blender, and (3) image synthesis through state-of-the-art generative AI tools. Each method contributes unique strengths in terms of authenticity, controllability, scalability, and diversity, which collectively help to mitigate the limitations associated with relying on a single data source.

Real-world images provide essential empirical grounding, reflecting the natural complexity of road scenes, including variable lighting, surface textures, weather effects, and occlusions. However, their collection is often limited by logistical, legal, and privacy constraints, as well as imbalanced representation of certain anomaly types. To overcome

[1] Yu, F. (undated). BDD100K. Available at: https://www.vis.xyz/bdd100k/ (Accessed: December 30, 2024).

these gaps, synthetic images generated in Blender offer complete control over scene parameters—such as lighting, camera angles, road textures, and object placement—enabling systematic dataset expansion and coverage of underrepresented or hazardous scenarios. Generative AI platforms, in turn, facilitate the creation of realistic images based on textual prompts, helping to simulate rare or visually complex anomalies that are difficult to obtain in either real-world or simulation-based data.

This section outlines the complete dataset development process employed in this study, detailing how each image source was utilized to cover three significant categories of road anomalies: **gravel**, **potholes**, and **puddles**. While the initial dataset construction focused on gravel anomaly detection and segmentation, the methodology was subsequently scaled to include the other two categories, enhancing the dataset's diversity and general applicability.

The final dataset comprises about 30,000 carefully curated and preprocessed annotated images designed to support deep learning models for object detection and segmentation. It serves not only as the foundation for the experimental framework of this research but also as a scalable and reusable resource for future studies in the domain of intelligent transportation systems.

Gravel images

To support the detection of gravel anomalies on roads, a dedicated subset of images was developed using a hybrid data generation approach. Due to the challenges of collecting large volumes of relevant real-world data—such as logistical constraints and limited availability of diverse gravel conditions—a combination of real, synthetic, and AI-generated images was employed to ensure sufficient variability and realism in gravel representations.

Real-world images. Gravel-related images were sourced from publicly available datasets and online image repositories featuring road surface anomalies. Only images clearly depicting gravel accumulations on roads were selected. These were manually curated based on relevance and visual clarity, and then annotated using the Roboflow platform, with bounding boxes marking the gravel areas.

Synthetic images (Blender). To expand the dataset with controllable and diverse gravel scenarios, synthetic scenes were created in Blender. Custom 3D road environments were designed, featuring varied lighting, asphalt textures, and gravel distributions. The Cycles rendering engine ensured high-quality outputs, and annotations were automatically generated using Blender's Python scripting tools. These images allowed systematic exploration of gravel types and environmental conditions, such as dry vs. wet roads or different gravel densities.

AI-generated images. Additional gravel imagery was produced using generative AI tools, including DALL·E and Adobe Firefly. Prompt engineering focused on generating complex or underrepresented gravel scenarios, such as gravel on road curves or in semi-urban settings. These images were manually reviewed for realism and labelled accordingly to ensure consistency with the rest of the dataset.

This focused image generation strategy helped create a diverse and representative gravel dataset subset, aimed at enhancing model robustness in detecting gravel anomalies under varied real-world and simulated conditions.

Real Images

Real images were sourced from public repositories and existing databases, ensuring the dataset's real-world representativeness. This data type is essential for grounding the dataset in authenticity and enabling models to learn the inherent characteristics of real-world imagery. However, the collection of real images can be limited by availability and variability constraints, necessitating integration with synthetic and AI-generated data. Figure 2 shows some examples of real-world images.

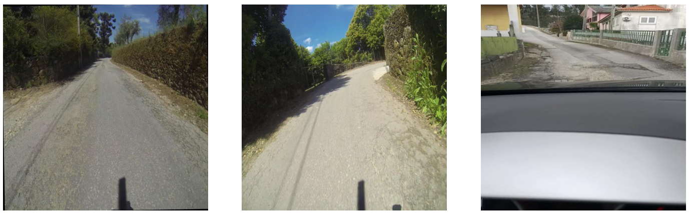

Figure 2. Examples of real-world images of road segments with gravel.

Augmented Images To address the limitations of relying solely on real-world imagery—such as restricted access to certain anomaly types, inconsistent lighting or weather conditions, and the labor-intensive nature of manual annotation—this study incorporated augmented images into the dataset. Augmentation was crucial not only for increasing the volume of available data but also for enhancing visual diversity, improving model generalization, and supporting training under edge-case scenarios.

Two complementary approaches were employed to generate these augmented images. The first relied on 3D simulation using Blender, a powerful and open-source 3D content creation suite. With Blender, it was possible to design controlled and repeatable environments, where parameters such as road texture, gravel distribution, lighting direction, camera angles, and weather conditions could be precisely manipulated. Moreover, Blender allowed for the automatic generation of annotations, significantly reducing the burden of manual labelling. This method proved especially valuable in replicating scenarios that are difficult, dangerous, or impractical to capture in real life—such as gravel on poorly lit mountain curves or wet roads after rainfall.

The second approach involved the use of Generative AI, leveraging models such as DALL·E, Adobe Firefly, and Stable Diffusion to synthesize high-quality, prompt-driven images. These tools enabled the rapid generation of realistic or stylized scenes by describing the desired anomaly and its context in natural language. For example, prompts were designed to produce images of gravel scattered across urban intersections or near construction zones, allowing the simulation of visually complex scenarios that rarely appear in public datasets. While these platforms do not natively produce annotations, their output was manually reviewed and labelled to ensure consistency with the rest of the dataset.

Together, these two augmentation strategies ensured a broad and realistic coverage of anomaly cases, reinforcing the dataset's ability to support robust, scalable, and transferable machine learning models for road anomaly detection.

Synthetic Data Generated in Blender

Blender, a 3D content creation tool, was utilized to generate synthetic image data relevant to the subject. This methodology enables the creation of controlled and reproducible scenarios, allowing for the precise adjustment of parameters such as lighting, camera angles, and textures. Synthetic data generation is particularly valuable for addressing gaps in real-world data and expanding the range of scenarios represented in the dataset.

Given the challenges associated with acquiring real-world data—including time constraints, legal considerations, and privacy concerns—Blender offers a viable alternative for generating synthetic data. It has gained prominence as an efficient and scalable solution. Among the available tools, Blender is distinguished by its ability to create photorealistic 3D scenarios with comprehensive control over lighting, textures, camera perspectives, and automatic annotations while also being a free and accessible resource.

The utilization of Blender for synthetic image generation is a significant asset for AI model training. Its ability to automate the creation of annotated datasets, coupled with the flexibility to simulate realistic environments, renders it an indispensable tool for researchers and industry professionals.

The future of this technology suggests the development of increasingly photorealistic synthetic datasets, combined with techniques such as domain adaptation and self-supervised learning, to ensure enhanced transferability to real-world data.

Synthetic data generation with Blender not only reduces costs and time but also expands the potential for innovation in computer vision, robotics, healthcare, and multimodal AI. Figure 3 shows some examples of images built using Blender [58–61].

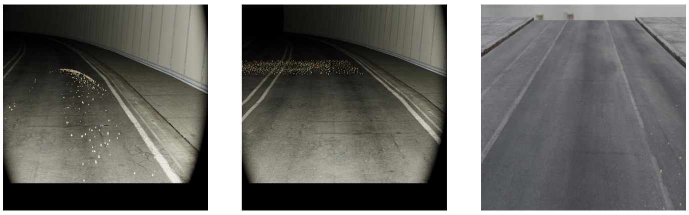

Figure 3. Examples of blender images of road segments with gravel.

Generative AI Images Recent advancements in generative AI techniques have facilitated the creation of supplementary data using GANs and diffusion models. These synthetic images augment dataset diversity by simulating rare or challenging-to-obtain real and synthetic data scenarios. Furthermore, AI-generated data has undergone rigorous validation to ensure its quality and relevance.

The creation of image datasets for AI models presents a critical challenge in fields such as computer vision, pattern recognition, and deep learning. Traditionally, these datasets consist of real-world images, but this approach suffers from significant limitations, including high data collection costs, privacy constraints, and limited availability of specific images. To address these challenges, generative AI models, such as GANs and their modern variants like StyleGAN and CycleGAN, have emerged as viable alternatives for generating realistic synthetic images [62, 63].

The use of generative AI for creating synthetic images has revolutionized dataset generation for deep learning models. Its ability to create highly detailed and annotated datasets while mitigating issues such as privacy, bias, and availability renders it an indispensable tool for researchers and industry professionals.

The future of this technology suggests the development of increasingly photorealistic synthetic datasets, coupled with techniques such as domain adaptation and self-supervised learning, to ensure enhanced generalization in the real world.

To enrich the dataset and ensure diversity across different types of road anomalies, this research employed a variety of state-of-the-art generative AI tools. Each platform offers distinct advantages in terms of image quality, customization, annotation support, and creative control. Below is an overview of the main technologies used in the image synthesis process:

- **Adobe Firefly**: Adobe Firefly is a generative AI tool integrated into the Adobe ecosystem, optimized for creative professionals. It supports high-resolution image generation with an emphasis on photorealism and artistic consistency. Firefly excels in style adherence and offers precise, prompt control with support for inpainting and background generation. Its integration with Photoshop and Adobe Express makes it highly suitable for workflows that require post-editing or multi-modal design outputs. An example of an image generated using Adobe Firefly is presented in Figure 4.

- **DALL·E (OpenAI)**: DALL·E is a powerful diffusion-based generative model that can produce high-fidelity and imaginative images from textual prompts. The model supports inpainting, outpainting, and prompt-based scene manipulation. Its strength lies in versatility and generalization across a wide range of visual domains. In this research, DALL·E was used to generate scenes of gravel, potholes, and puddles with varied weather, lighting, and material textures. An example of an image generated using Dall-E is presented in Figure 5.

- **Playground AI**: Playground AI is an accessible web-based interface that supports models like Stable Diffusion and DALL·E, offering fine-grained prompt control and batch image generation. It features strong community integration, allowing users to explore pre-configured styles and replicate visual templates. Its flexibility and ease of use made it suitable for rapid prototyping and testing multiple scene variations within road anomaly contexts. An example of an image generated using Playground is presented in Figure 6.

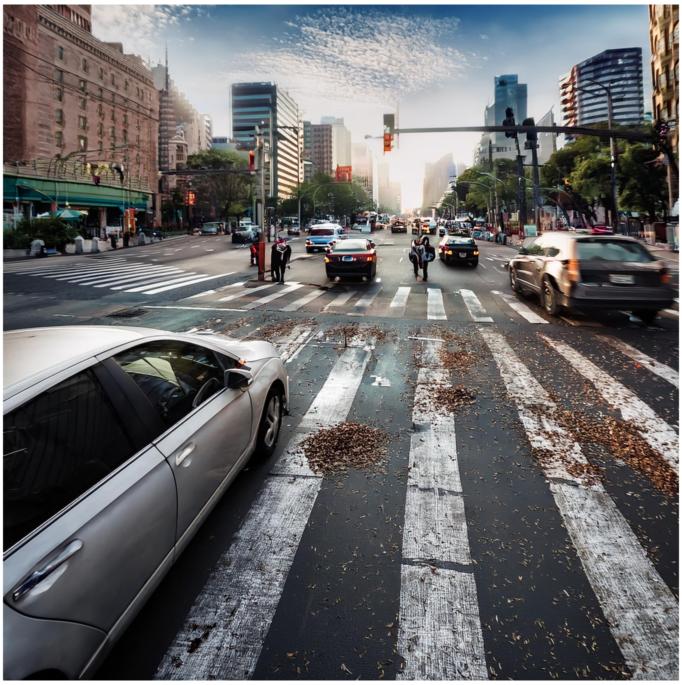

Figure 4. Example of image generated using Adobe Firefly.

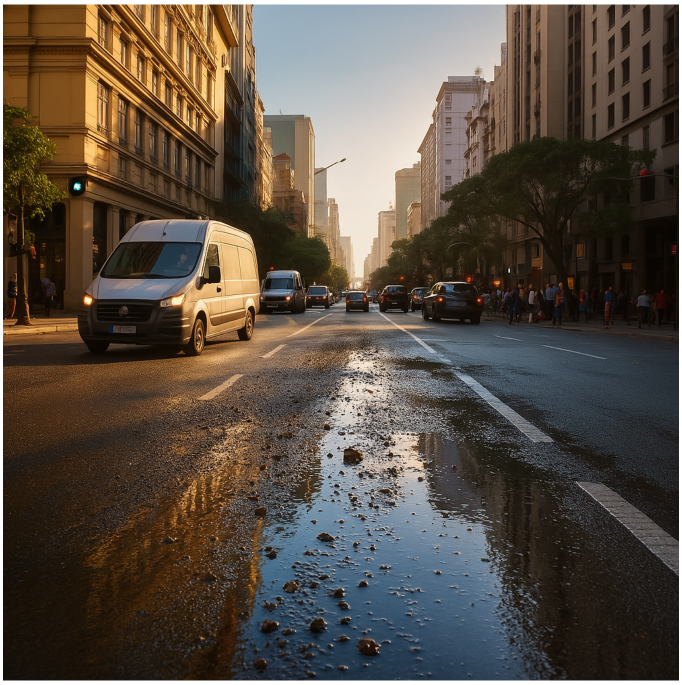

Figure 5. Example of image generated using DALL·E.

- **Adobe Express (with Firefly integration)**: Adobe Express combines design layout tools with

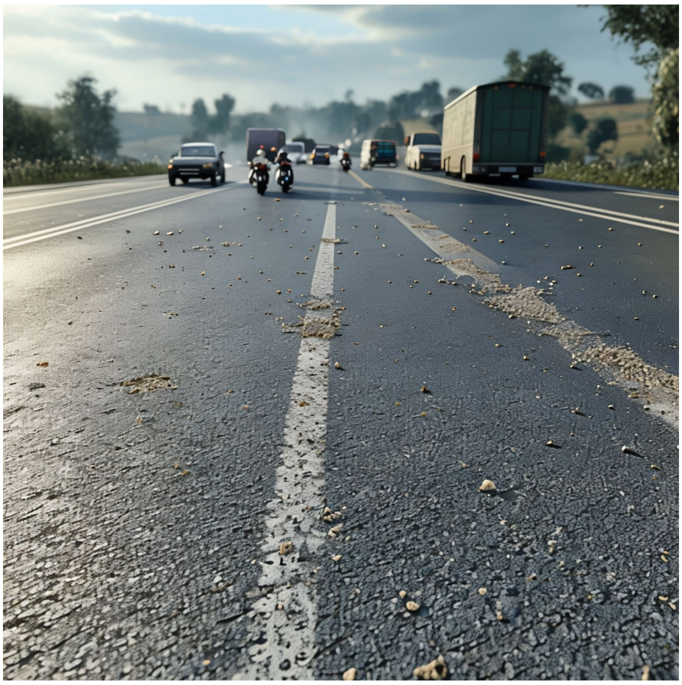

Figure 6. Example of image generated using Playground AI.

Firefly-powered image generation. While less technical than Photoshop, it provides intuitive controls for style selection, content composition, and prompt variation. Its role in this project was primarily to create contextually rich backgrounds and compositions for composite anomaly scenes. An example of an image generated using Adobe Express is presented in Figure 7.

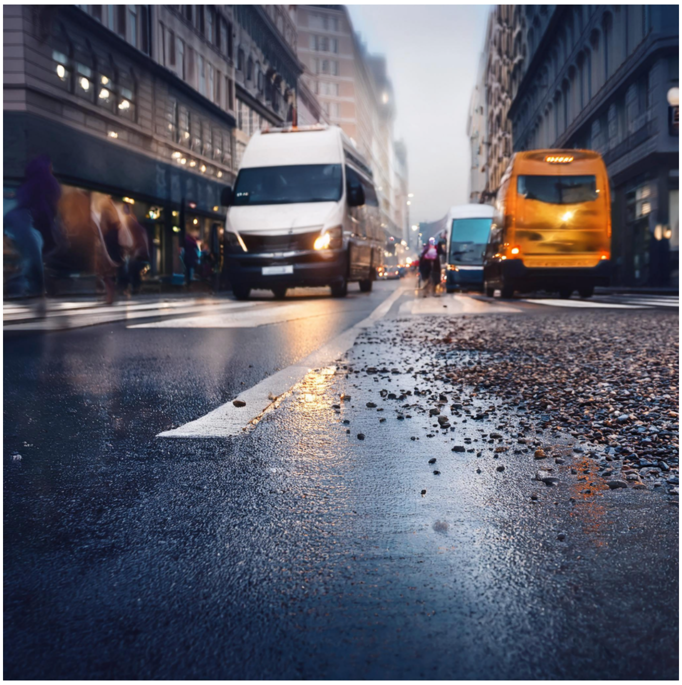

Figure 7. Example of image generated using Adobe Express.

Each generative AI tool used in this research was evaluated based on key criteria, including usability, image realism, control over environmental variables, and output consistency. The hybrid adoption of multiple platforms proved beneficial, as it enabled the generation of images under varied lighting conditions, weather conditions, and road conditions. It also allowed the simulation of both common and rare anomaly scenarios—including gravel, potholes, and puddles—thus enhancing dataset diversity and representativeness. The combination of photorealistic and semi-stylized renderings contributed to greater domain robustness. At the same time, the mix of high-end tools (e.g., Adobe Firefly, DALL·E) with high-throughput generators (e.g., Playground AI, Easy Diffusion) provided an effective balance between quality and volume.

Despite their strengths, generative AI tools present certain limitations. Many models are susceptible to prompt phrasing, requiring multiple iterations to produce images with the desired realism and contextual accuracy. Additionally, unlike simulation-based tools such as Blender, these platforms typically lack native support for semantic annotations, which often necessitates manual labeling or the use of external tools for segmentation and tagging. Some generated images may contain visual artifacts, such as distorted geometries or unnatural reflections, which must be filtered or corrected through post-processing. Moreover, output style consistency can vary significantly between tools due to differences in training data and model architectures. Another significant limitation is the cost barrier: many of the most capable generative platforms operate under subscription-based or pay-per-use models, which can restrict access for individual researchers or institutions with limited budgets, thereby impacting reproducibility and scalability.

In summary, the integration of diverse generative AI platforms enabled the creation of a visually rich and representative dataset, essential for the robust training of machine learning models in road anomaly detection. These tools provided a scalable and flexible approach to overcoming the challenges of real-world data acquisition, enabling the simulation of rare or hazardous conditions with high configurability and creative control.

Pothole Image

Potholes are among the most prevalent and dangerous types of road surface anomalies. They are typically caused by the erosion of the pavement surface due to weather conditions, water infiltration, and repeated mechanical stress. In this study, real-world images of potholes were collected from publicly available datasets and repositories to ensure authenticity and ensure the accuracy of ground truth.

The use of real images is crucial for training models that can be generalized effectively in real-world driving conditions. These images exhibit natural variability in terms of lighting, road texture, pothole shape, and surrounding context—factors that are difficult to simulate with high fidelity in synthetic environments.

Pothole images were sourced from datasets such as *Pothole Dataset 1* available on Roboflow at https://universe.roboflow.com/testing-6yjel/pothole-detection-yolov12/images/1brzinWTx1x6wPBaahWF, *Pothole Dataset 2* available on Roboflow at https://universe.roboflow.com/aegis/pothole-detection-i00zy/dataset/3, and *Pothole Dataset 3* available on Roboflow at https://public.roboflow.com/object-detection/pothole/1, as well as various road maintenance archives. Each image underwent preprocessing and annotation to ensure consistency across the dataset.

Examples of these images are shown in Figure 8, depicting road segments with different pothole characteristics, including size, shape, and illumination conditions.

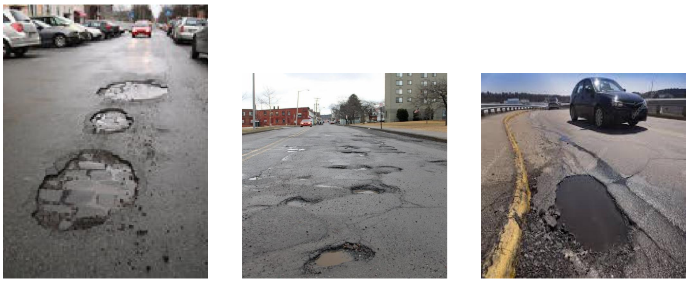

Figure 8. Examples of real-world images of road segments with potholes.

Puddle Image

Puddles represent another common anomaly found on roads, particularly after rainfall or in poorly drained areas. Although often underestimated, puddles pose significant safety risks due to hidden surface defects, reduced tire grip, and the potential for hydroplaning. Moreover, water accumulation may obscure potholes or road cracks, complicating the detection of anomalies.

Real-world puddle images were sourced from open-access image datasets, including traffic surveillance archives and autonomous vehicle research datasets. Specifically, the datasets used include the *Puddle Dataset 1*, available on Roboflow at https://universe.roboflow.com/hanyang-university-bd2kb/puddle-detection/dataset/8; the *Puddle Dataset 2* dataset, also hosted on Roboflow at https://universe.roboflow.com/yay-gdjrj/-Ic0sk/dataset/1; and the *Puddle Dataset 3*, available on Kaggle at https://www.kaggle.com/datasets/carolineadicahya04/puddle. These datasets were selected to cover a wide variety of conditions and scenarios.

The images in these datasets were chosen to represent a range of conditions, including daylight reflections, shadow interference, and various road textures.

As shown in Figure 9, puddle scenarios vary widely in appearance depending on light incidence, background reflectivity, and water depth. These examples underscore the importance of high dataset variability in training robust detection systems.

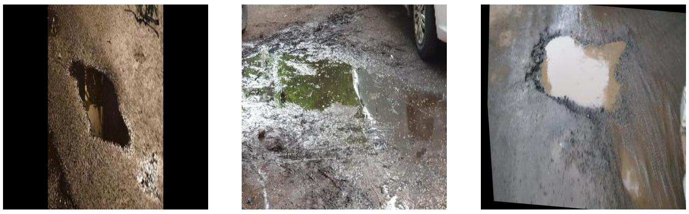

Figure 9. Examples of real-world images of road segments with puddles.

3.3.3 Dataset Summary

The final dataset developed in this research comprises a total of **28,764 images**, distributed across three primary road anomaly categories: *gravel*, *potholes*, and *puddles*. Each anomaly type is represented by a combination of real-world, synthetic (Blender), and AI-generated images, ensuring diversity, realism, and

Table 2. Balanced dataset composition by anomaly type and image source.

Anomaly Type	Total Used	Real-World	Synthetic (Blender)	AI-Generated
Gravel	8,304	1,362	5,471	1,471
Potholes	10,200	10,200	-	-
Puddles	10,260	10,260	-	-
Total	**28,764**			

generalizability. Table 2 summarizes the dataset composition.

Initially, the available dataset included up to **14,000 images** for each of the *pothole* and *puddle* categories, and **8,304 images** for the *gravel* category. However, to promote better class balance and enhance training robustness, the number of *pothole* images was intentionally reduced to **10,200**, selected based on quality and variety criteria. This adjustment resulted in a more balanced and representative dataset across all categories.

It is worth noting that, within the *gravel* class, there is a noticeable discrepancy between the number of real-world, synthetic, and AI-generated images. Despite this imbalance, the visual similarity between the real and AI-generated gravel images helps to mitigate its impact. Furthermore, the use of Blender for synthetic data generation allows for the rapid creation of hundreds of diverse images in a matter of minutes, making it feasible to populate the dataset with a large volume of synthetic samples. This efficiency contributed significantly to the higher proportion of synthetic images in the gravel category without compromising variability or quality.

All images were preprocessed to ensure uniform resolution, aspect ratio, and annotation format, enabling seamless integration into deep learning pipelines. The hybrid composition of the dataset combining empirical, simulated, and AI-generated data serves as a strong foundation for developing and testing models in road anomaly detection, capable of operating under real-world variability and edge-case conditions.

4 Hardware

This section provides an overview of the edge computing devices used in this study and discusses the rationale for selecting the corresponding object detection models. The goal is to evaluate the suitability of lightweight AI models for real-time inference in resource-constrained embedded environments.

The *Raspberry Pi 4* is a compact and cost-effective single-board computer equipped with a quad-core ARM Cortex-A72 processor (1.5 GHz), a Broadcom VideoCore VI GPU, and up to 8 GB of LPDDR4 RAM. In this study, the 4 GB variant was used to evaluate model performance under limited computational resources [64].

The *Raspberry Pi 5* represents the latest generation of the platform, featuring a Broadcom BCM2712 2.4 GHz quad-core 64-bit Arm Cortex-A76 CPU with cryptography extensions, 512 KB per-core L2 caches, and a 2 MB shared L3 cache. The 8 GB variant was selected to assess the potential gains in inference performance compared with its predecessor [65].

The *Jetson Nano 4 GB* (with JetPack) is a small yet powerful edge-AI platform developed by NVIDIA. It integrates a quad-core ARM Cortex-A57 CPU at 1.43 GHz and a 128-core Maxwell GPU, making it suitable for parallel computing tasks such as image recognition and object detection. With 4 GB of LPDDR4 RAM and support for the JetPack SDK running on Ubuntu, the device provides a complete AI development environment and serves as a reference for GPU-based embedded inference [66].

The models evaluated in this work were selected to cover a wide spectrum of architectural paradigms and to compare their accuracy–latency trade-offs on the aforementioned hardware platforms. Specifically, the YOLO family (v5, v8, v11, and v12) was chosen for its proven balance between detection accuracy and inference speed, with lightweight variants (`nano` and `small`) optimized for ARM and GPU-based deployment. RT-DETRv2 was included as a transformer-based model representing the current state of the art in detection accuracy, while *SSDLite* served as a classical mobile baseline built upon MobileNetV2 for comparison with modern architectures.

The primary focus of the experiments was the evaluation of **inference time and computational feasibility** on edge devices, as these metrics directly determine the real-time applicability of object detection models in embedded environments. Accuracy was considered an essential complementary metric; however, achieving consistent detection at low latency under constrained hardware conditions was the determining factor in assessing each model's suitability for real-world ARAS and ADAS scenarios.

It should be noted that some experimental limitations arose due to hardware availability and lack of on-vehicle testing equipment. Consequently, full validation under adverse conditions—such as nighttime illumination or wet-road surfaces—was not possible. Nevertheless, inference benchmarking and accuracy assessments were rigorously conducted under controlled conditions using curated datasets, providing reliable insights into each model's real-time performance potential.

Overall, the selected hardware platforms and model configurations encompass the main spectrum of efficiency–accuracy trade-offs relevant to embedded AI applications. The analysis presented in the subsequent sections directly builds upon these findings, linking device performance to the broader objective of enabling low-cost, real-time rider assistance systems.

5 Results and Discussion

5.1 Experimental Results

5.1.1 Overview

The models evaluated comprise representative architectures from diverse design paradigms, including **single-shot detectors** like SSD, multiple generations of the **YOLO family** (from YOLOv3-Tiny to YOLOv12), and **transformer-based models** such as RT-DETRv2. These selections reflect their prominence in the field of computer vision, widespread community adoption, and demonstrated effectiveness in real-time object detection tasks.

A central aspect of this study involves benchmarking these models on embedded platforms, specifically the Raspberry Pi 4, Raspberry Pi 5, and NVIDIA Jetson Nano. These devices exemplify the computational constraints inherent in real-world deployments of intelligent systems, especially in automotive contexts. Each model was tested using a custom dataset derived from the large-scale BDD100K database, tailored to encompass 16 distinct real-world driving conditions.

This chapter is structured as follows: The first section provides a comparative analysis of model performance on edge devices, covering metrics such as accuracy, precision, recall, F1-score, and inference latency. Special emphasis is placed on the inherent trade-offs between computational cost and detection effectiveness. Inference was performed using a curated subset of the BDD100K dataset, a large-scale driving video dataset containing annotated images from diverse road, weather, and lighting conditions, to ensure realistic and varied testing scenarios relevant to autonomous driving.

The second section offers a detailed performance analysis of YOLO models specifically for road anomaly detection, targeting the classes `gravel`, `pothole`, and `puddle`. This analysis employs both metric-based evaluation and diagnostic curve analysis, including an examination of the evolution of loss functions, average precision metrics, and class-wise performance trends. This comprehensive approach provides deeper insights into model behavior under operational constraints.

5.1.2 Edge-Device Benchmarking

This section details the evaluation of various object detection models that perform inference on resource-constrained edge devices, including the Raspberry Pi 4, Raspberry Pi 5, and Jetson Nano. The primary goal was to assess the trade-offs between model complexity, inference speed (latency), and detection accuracy across these platforms in scenarios relevant to autonomous driving. A customized dataset, derived from the large-scale BDD100K database[2] and categorized into 16 distinct real-world driving conditions, was utilized for these inference tests.

The results obtained from these inference experiments are reported in the study by Santos et al. [67], which evaluated the customized dataset derived from BDD100K across multiple edge platforms. That work presents a detailed comparison of accuracy, precision, recall,and F1-score at two different image resolutions. The findings indicate that higher image resolutions generally improve detection performance,at the expense of increased computational load. Furthermore, lightweight models such as YOLOv8n and YOLOv9t were identified as strong candidates for real-time deployment, as they provide a balanced compromise between inference speed and detection accuracy on resource-constrained devices.

Among the tested models, RT-DETRv2 (r34vd) achieved the highest accuracy and F1-score, highlighting its potential for demanding detection tasks. However, its computational requirements make it less suitable for real-time operation on devices with limited resources. Conversely, models such as YOLOv5n, YOLOv8n, and YOLOv11n provided significantly faster inference times while maintaining respectable detection performance, which makes them strong candidates for embedded vision systems where real-time decision-making is essential, such as in autonomous driving.

Inference Performance on Edge Devices Inference tests were performed on three edge devices—**Raspberry Pi 4**, **Raspberry Pi 5**, and **Jetson Nano**—to evaluate real-world performance of various object detection models under constrained hardware conditions. Results indicate a clear trade-off between

[2] Yu, F. (undated) BDD100K. Available at: https://www.vis.xyz/bdd100k/ (Accessed: December 30, 2024)

model complexity and inference speed. Lightweight models such as **YOLOv5n** and **YOLOv8n** achieve near real-time inference on all devices at 320×320 resolution, while larger transformer-based models like **RT-DETRv2 (r34vd)** show significantly higher latencies, particularly on Raspberry Pi 4, making them impractical for real-time applications on resource-limited hardware. Raspberry Pi 5 generally improves inference times compared to Raspberry Pi 4, while the Jetson Nano offers competitive performance for mid-sized models but is limited for some transformer-based architectures due to software compatibility constraints. This results are shown on Table 3.

5.1.3 *YOLO-Based Road Anomaly Detection*

This subsection presents the experimental evaluation conducted to assess the performance of state-of-the-art YOLO-based object detection models for road anomaly detection. The experiments were designed to investigate the effectiveness of different architectures (YOLOv5, YOLO11, and YOLO12), model scales (nano and small variants), and input resolutions (320×320 and 640×640 pixels). The analysis covers both quantitative metrics, such as mAP, F1-score, and class-specific average precision, as well as qualitative inference results illustrated through representative detection examples. The goal of this evaluation is to identify the most suitable models for deployment on resource-constrained edge devices, balancing accuracy, robustness, and computational efficiency for real-time road anomaly detection.

Model Performance Evaluation We evaluated six YOLO model variants (YOLOv5n, YOLOv5s, YOLO11n, YOLO11s, YOLO12n, YOLO12s) for road anomaly detection across three classes: `gravel`, `pothole`, and `puddle`. Each model was trained at two input resolutions (320×320 and 640×640 pixels) for 270–300 epochs.

Table 4 presents the comprehensive performance comparison across all evaluated models. The YOLO11 and YOLO12 series achieved superior performance, with mAP@0.5 values ranging from 0.853 to 0.872. Notably, `gravel` consistently achieved the highest detection accuracy across all models ($\text{AP}_{0.5} > 0.92$), while `pothole` remained the most challenging class due to high intra-class variability.

Qualitative Inference Examples

5.2 Analysis

5.2.1 *Synthesis of Edge-Device Inference Findings*

The inference results across the three edge devices highlight the crucial trade-offs between model complexity, computational resources, and real-time performance for object detection in autonomous driving scenarios. Lightweight models, notably smaller variants of YOLO, consistently demonstrate faster inference times across all platforms, making them more suitable for applications that demand low latency. The Raspberry Pi 5 demonstrated significant improvements in inference speed compared to the Raspberry Pi 4, enabling the deployment of slightly more complex models while maintaining reasonable frame rates. The Jetson Nano, with its dedicated GPU, offered a competitive performance profile, excelling with specific model architectures but also exhibiting limitations with larger, more computationally intensive networks.

The choice of the optimal model for edge deployment ultimately depends on the specific requirements of the application, balancing the need for high detection accuracy with the constraints of the target hardware and the criticality of real-time processing. Further research could explore model optimization techniques, such as quantization and pruning, to enhance the efficiency of larger models on these resource-constrained platforms. Additionally, evaluating the energy consumption of these models during inference would be crucial for real-world automotive applications. Among all the evaluated models, **YOLOv5nu** and **YOLOv11n** demonstrated the most consistent results across all hardware platforms, offering a good balance between detection performance and low inference time. However, the **best overall model** for edge deployment is **YOLOv11n running on the Jetson Nano** with an image size of 320×320, as it achieved both fast inference times and acceptable detection accuracy, making it the most suitable option for real-time applications in resource-constrained environments.

5.2.2 *Performance Evaluation and Interpretation for Road Anomalies*

From a class-specific perspective, performance varied considerably across anomaly categories. `Gravel` consistently achieved the highest $\text{AP}_{0.5}$ scores (0.92–0.95), benefiting from stable texture patterns and robust feature representation. `Puddle` detection obtained moderate results (0.79–0.84), with performance often degrading at higher recall levels

Table 3. Inference times (s) on edge devices for all tested models.

Model	Parameters	Image Size	Raspberry Pi 4	Raspberry Pi 5	Jetson Nano
SSDLite	3.4M	320×320	0.48	0.151	0.213
		640×640	0.49	0.153	0.211
YOLOv3-Tiny	8.7M	320×320	0.348	0.133	0.060
		640×640	1.118	0.346	0.185
YOLOv5n	1.9M	320×320	0.183	0.060	0.068
		640×640	0.589	0.220	0.225
YOLOv5s	7.5M	320×320	0.413	0.161	0.092
		640×640	1.478	1.466	0.246
YOLOv5u	7.5M	320×320	0.413	0.161	0.092
		640×640	1.478	1.466	0.246
YOLOv7-Tiny	6.2M	320×320	0.249	0.065	0.292
		640×640	0.911	0.274	1.108
YOLOv8n	3.2M	320×320	0.203	0.063	0.041
		640×640	0.637	0.290	0.132
YOLOv8s	11.2M	320×320	0.462	0.152	0.128
		640×640	1.700	0.653	0.269
YOLOv9t	2M	320×320	0.217	0.068	0.137
		640×640	0.702	0.249	0.156
YOLOv9s	7.2M	320×320	0.496	0.153	0.357
		640×640	1.800	0.553	0.307
YOLOv10n	2.3M	320×320	0.213	0.071	0.044
		640×640	0.671	0.253	0.129
YOLOv10s	7.2M	320×320	0.482	0.175	0.084
		640×640	1.671	0.512	0.268
YOLOv11n	2.6M	320×320	0.185	0.062	0.045
		640×640	0.562	0.264	0.130
YOLOv11s	9.4M	320×320	0.414	0.144	0.082
		640×640	1.565	0.619	0.283
YOLOv12n	2.5M	320×320	0.186	0.083	0.055
		640×640	0.659	0.352	0.173
YOLOv12s	9.1M	320×320	0.441	0.162	0.094
		640×640	1.866	0.763	0.647
RT-DETRv2(rv18d)	20M	320×320	3.498	–	–
		640×640	3.498	0.928	–
RT-DETRv2(r34vd)	31M	320×320	5.228	–	–
		640×640	5.228	1.300	–

due to the challenges posed by reflective surfaces and varying lighting conditions. The `Pothole` class proved to be the most challenging (0.76–0.84), reflecting the irregular shapes and high variability in appearance, which complicated robust feature extraction.

Resolution analysis confirmed that increasing the input size from 320×320 to 640×640 pixels generally improved detection accuracy, with mAP@0.5 gains of approximately 1–4% across models. However, these improvements were modest, suggesting that 320×320 resolution remains the most practical compromise between computational efficiency and detection accuracy, especially for real-time applications on edge devices.

When comparing architectures, the YOLO11 series demonstrated the best overall performance, with YOLO11n achieving 0.872 mAP@0.5 at 320×320 while maintaining strong computational efficiency. YOLO12 models showed competitive behavior, particularly enhancing performance in pothole detection. Although YOLOv5 variants obtained slightly lower scores overall, they proved to be highly

Table 4. Performance comparison of YOLO models for road anomaly detection.

Model	Resolution	mAP@0.5	Recall	Max F1	Threshold
YOLOv5n	320×320	0.832	0.940	0.79	0.235
YOLOv5n	640×640	0.848	0.930	0.81	0.349
YOLOv5s	320×320	0.853	0.930	0.80	0.344
YOLOv5s	640×640	0.858	0.920	0.82	0.344
YOLO11n	320×320	0.872	0.920	0.82	0.310
YOLO11n	640×640	0.861	0.930	0.82	0.345
YOLO11s	320×320	0.853	0.930	0.83	0.340
YOLO11s	640×640	0.865	0.910	0.82	0.326
YOLO12n	320×320	0.834	0.940	0.80	0.331
YOLO12n	640×640	0.871	0.940	0.82	0.308
YOLO12s	320×320	0.857	0.920	0.82	0.310
YOLO12s	640×640	0.872	0.920	0.82	0.310

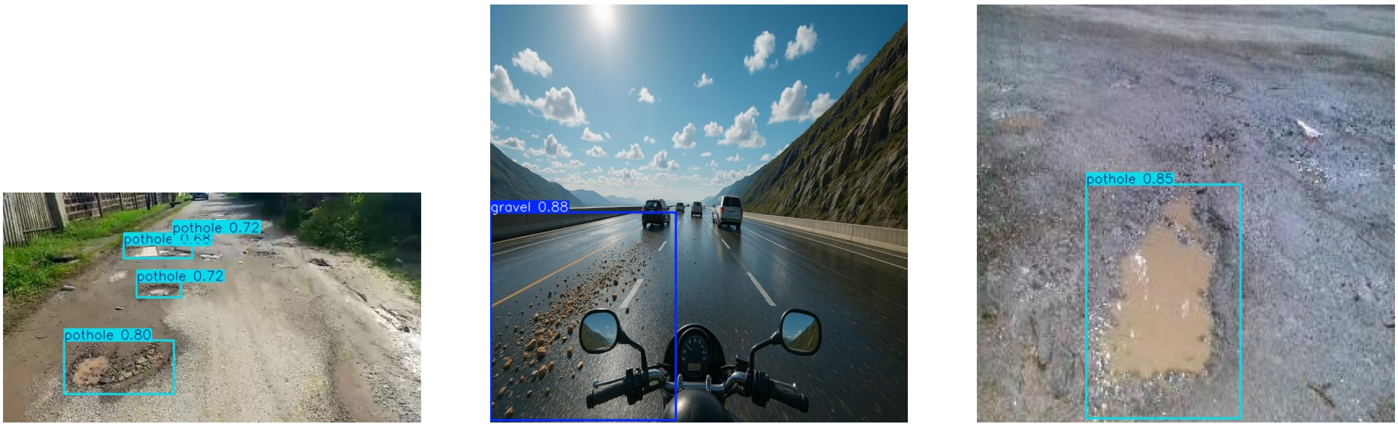

Figure 10. Inference results from YOLOv11n model trained with 320×320 resolution.

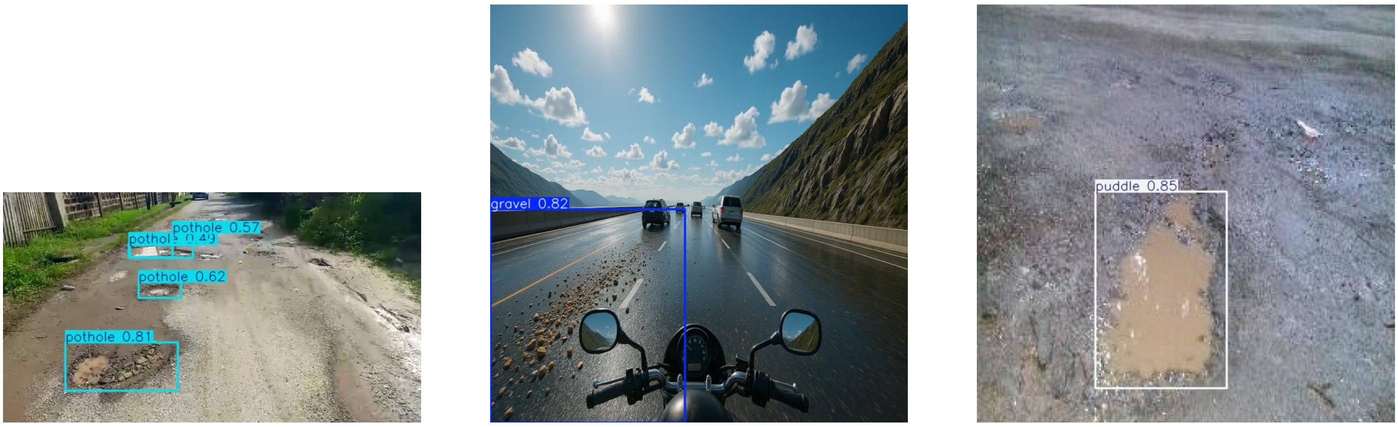

Figure 11. Inference results from YOLOv12n model trained with 320×320 resolution.

stable and reliable, making them attractive options for deployments constrained by limited hardware resources.

The qualitative inference analysis confirmed the quantitative findings. As shown in Figures 10 and 11, all models successfully detected potholes, gravel, and puddles across diverse real-world scenarios, with confidence scores typically ranging from 0.62 to 0.88 and reasonably placed bounding boxes under varying lighting and road conditions. Moreover, inference with 320×320 resolution consistently resulted in tighter bounding boxes and improved spatial precision, reinforcing the observed trade-off between accuracy and computational cost.

5.2.3 Discussion and Implications

This study presents a comprehensive evaluation of deep learning-based road anomaly detection systems specifically designed for motorcycle-oriented ARAS applications. The experimental results are not only quantitative benchmarks but also provide meaningful insights into the trade-offs between model accuracy, latency, and feasibility for embedded deployment. The primary purpose of this work was to assess how well state-of-the-art detection architectures can operate under the computational and memory constraints of affordable edge devices, such as the Raspberry Pi4, Raspberry Pi5, and Jetson Nano.

The results clearly demonstrate that performance differences among models are consistent with their architectural design principles. Transformer-based detectors, such as RT-DETRv2, achieved the highest detection accuracy and F1-scores, confirming their superior representational capacity. However, their high computational cost and memory footprint made them unsuitable for real-time inference on the Raspberry Pi platforms. In contrast, the lightweight YOLO variants (particularly YOLOv5n, YOLOv8n, and YOLOv11n) achieved near real-time inference speeds while maintaining competitive accuracy levels, proving to be the most balanced choices for embedded motorcycle applications. These findings substantiate the conclusion that, in resource-constrained environments, inference latency and energy efficiency must take precedence over marginal gains in accuracy.

The benchmarking of inference times across hardware platforms is a central contribution of this study and directly supports the discussion. The comparative evaluation showed that the Raspberry Pi5 achieves up to 40–50% faster inference than the Raspberry Pi4, while the Jetson Nano benefits from GPU acceleration but faces software compatibility limitations for transformer-based models. This systematic comparison provides practical guidance for developers seeking to select the most appropriate model–hardware combination for real-time road anomaly detection in ARAS systems.

Furthermore, the results highlight the importance of dataset diversity and model scalability. Models trained with the hybrid dataset (real, synthetic, and AI-generated images) demonstrated robust generalization across gravel, pothole, and puddle classes, confirming the validity of the adopted data generation strategy. The consistent performance of YOLOv11n at 320×320 resolution shows that reduced image sizes can achieve adequate detection while drastically lowering inference time and power consumption—a key insight for embedded system design.

It is also important to note that some experiments could not be fully completed due to the limited availability of testing equipment and environmental capture devices. The hardware resources available during this study did not allow for extensive real-world data collection under varying conditions (e.g., night or rain). Nevertheless, the obtained results are sufficiently comprehensive to establish the performance boundaries of each model on the selected edge platforms, which was the principal focus of this work.

In summary, the discussion establishes a clear relationship between the experimental findings and the study's objectives. The results confirm that lightweight YOLO architectures offer the most practical solution for real-time deployment in motorcycle ARAS systems, balancing inference speed, detection accuracy, and hardware efficiency. These insights provide an evidence-based foundation for future research, particularly for extending the experiments to on-vehicle validation under diverse environmental conditions and exploring further optimization techniques such as quantization, pruning, and sensor fusion.

6 Conclusion

This work demonstrates the technical and practical feasibility of deploying deep learning-based road anomaly detection systems on embedded platforms for motorcycle safety applications. The comprehensive evaluation across multiple YOLO architectures, input resolutions, and hardware platforms provides valuable insights for system designers and researchers working on similar applications.

This work provides a cohesive and practical framework that advances the state of the art in road anomaly detection for motorcycle safety. By integrating hybrid dataset generation, multi-model benchmarking, and edge-oriented deployment analysis, it directly addresses the key research gaps identified in the introduction.

The main **validated contributions** are:

- The **hybrid dataset strategy** effectively mitigated class imbalance and scarcity issues, enabling robust detection across diverse conditions.

- The **comparative model–hardware benchmark** established the performance boundaries of modern object detectors on low-cost embedded platforms, revealing that YOLOv11n offers the best balance of accuracy and efficiency.
- The **edge-optimization analysis** confirmed that real-time inference is achievable on devices such as the Jetson Nano and Raspberry Pi 5, defining practical configurations for future ARAS integration.
- The **deployment insights and guidelines** provide a transferable framework for scaling AI-based rider assistance systems across platforms and use cases.

The YOLO11n model trained at 320×320 resolution emerged as the optimal solution, achieving 0.872 mAP@0.5 while maintaining real-time inference capabilities on affordable edge devices. This represents a significant step toward democratizing safety-critical technologies for motorcyclists, offering potential for broad adoption of AI-powered rider assistance systems.

Future work should prioritize field validation under real riding conditions, integration of multi-modal sensing capabilities, development of adaptive learning strategies for personalized detection thresholds, and optimization of energy efficiency through model quantization and pruning techniques. The temporal modeling of video sequences and development of integrated warning systems represent additional promising research directions.

This research establishes a strong foundation for advancing motorcycle safety through intelligent transportation systems, combining technical innovation with practical deployment considerations to address real-world safety challenges faced by vulnerable road users. The demonstrated feasibility of high-performance, low-cost detection systems opens new possibilities for inclusive and accessible advanced rider assistance technologies.

Data Availability Statement

Data will be made available on request.

Funding

The study was developed under the project A-MoVeR – "Mobilizing Agenda for the Development of Products & Systems towards an Intelligent and Green Mobility", operation n.º 02/C05-i01.01/2022.PC646908627-00000069, approved under the terms of the call n.º 02/C05-i01/2022 – Mobilizing Agendas for Business Innovation, financed by European funds provided to Portugal by the Recovery and Resilience Plan (RRP), in the scope of the European Recovery and Resilience Facility (RRF), framed in the Next Generation UE, for the period from 2021-2026.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Kozicki, B., & Skrabacz, A. (2024). A comparative analysis of injuries and deaths caused by road traffic accidents in Poland and selected EU countries. *Transport Problems, 19*. [CrossRef]

[2] Nycz, M., & Sobolewski, M. (2025). The Problem of the Comparability of Road Accident Data from Different European Countries. *Sustainability, 17*(15), 6754. [CrossRef]

[3] Wu, D., Hours, M., & Martin, J. L. (2018). Risk factors for motorcycle loss-of-control crashes. *Traffic injury prevention, 19*(4), 433-439. [CrossRef]

[4] Lyu, H., Wang, Z., Lin, P.-S., Lu, Q., Duran, E., & Hsu, P. P. (2025). How Does Pavement Friction Affect Motorcycle Crashes? A Florida Analysis. *Transportation Research Record, 2679*(10), 647–661. [CrossRef]

[5] Hossain, S., Maggi, E., & Vezzulli, A. (2024). Factors influencing the road accidents in low and middle-income countries: a systematic literature review. *International Journal of Injury Control and Safety Promotion, 31*(2), 294–322. [CrossRef]

[6] Chand, A., Jayesh, S., & Bhasi, A. B. (2021). Road traffic accidents: An overview of data sources, analysis techniques and contributing factors. *Materials Today: Proceedings, 47*, 5135–5141. [CrossRef]

[7] Zhao, T., He, J., Lv, J., Min, D., & Wei, Y. (2023). A Comprehensive Implementation of Road Surface Classification for Vehicle Driving Assistance: Dataset, Models, and Deployment. *IEEE Transactions on Intelligent Transportation Systems, 24*(8), 8361–8370. [CrossRef]

[8] Saeed, N., Nyberg, R. G., & Alam, M. (2023). Gravel road classification based on loose gravel using transfer learning. *International Journal of Pavement Engineering, 24*(2), 2138879. [CrossRef]

[9] Saeed, N., Alam, M., & Nyberg, R. G. (2024). A multimodal deep learning approach for gravel road condition evaluation through image and audio integration. *Transportation Engineering, 16*, 100228. [CrossRef]

[10] Ullah, Z., da Silva, J. A., Nunes, R. R., Reis, A., Filipe, V., Barroso, J., & Pires, E. J. (2025). Performance of Advanced Rider Assistance Systems in Varying Weather Conditions. *Vehicles (2624-8921), 7*(4). [CrossRef]

[11] Ali, W. A. E., Saleh, H. A., Eid, H. S., Fathi, A. M., & Mohamed, A. E. R. A. (2024, July). Design and implementation of advanced driver assistance systems. In *2024 International Telecommunications Conference (ITC-Egypt)* (pp. 1-6). IEEE. [CrossRef]

[12] Ahmad Jan, M., Adil, M., Brik, B., Harous, S., & Abbas, S. (2025). Making Sense of Big Data in Intelligent Transportation Systems: Current Trends, Challenges and Future Directions. *ACM Computing Surveys, 57*(8), 1-43. [CrossRef]

[13] Li, G., Zhang, C., Li, M., Han, D. L., & Zhou, M. L. (2024). LHA-Net: a lightweight and high-accuracy network for road surface defect detection. *IEEE Transactions on Intelligent Vehicles*. [CrossRef]

[14] Zhao, L., & Zhu, M. (2023). MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography. *Drones, 7*(3), 188. [CrossRef]

[15] Chen, D. R., & Chiu, W. M. (2020, November). Road crack detection using Gaussian mixture model for diverse illumination images. In *2020 30th International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 1-6). IEEE. [CrossRef]

[16] Zhou, W., Zhan, Y., Zhang, H., Zhao, L., & Wang, C. (2022). Road defect detection from on-board cameras with scarce and cross-domain data. *Automation in Construction, 144*, 104628. [CrossRef]

[17] Zhang, R., Yang, S., Lyu, D., Wang, Z., Chen, J., Ren, Y., Gao, B., & Lv, Z. (2025). AGSENet: A Robust Road Ponding Detection Method for Proactive Traffic Safety. *IEEE Transactions on Intelligent Transportation Systems, 26*(1), 497–516. [CrossRef]

[18] Chang, J.-Y., & Xu, Z.-X. (2025). Enhanced Water Puddle Segmentation and Detection Using DCU-Net. *IEEE Internet of Things Journal*, 12(1), 663–672. [CrossRef]

[19] Hasanaath, A. A., Mohammed, A. S., Alotaibi, L., Alabdulrahman, M., & Moinuddeen, A. (2024). Pothole Severity Classification Using Structural Features Extracted from Segmented Images. In *2024 Arab ICT Conference (AICTC)* (pp. 52–58). [CrossRef]

[20] Sun, Z., Zhu, L., Qin, S., Yu, Y., Ju, R., & Li, Q. (2024). Road Surface Defect Detection Algorithm Based on YOLOv8. *Electronics*, 13(12), 2413. [CrossRef]

[21] Pratheesh Raj, P. L., Santhiya, M., Stewart Kirubakaran, S., Leelipushpam, G. J., Ebenezer, V., & Bijolin Edwin, E. (2024). Pothole Detection and Alert System Sing YOLO Models: A Real-Time Driver Assistance Approach. In *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)* (pp. 1555–1559). [CrossRef]

[22] Amer, R. T., Soubra, H., & Mazhr, A. (2024). Advanced Autonomous Rider Assistance for Smart E-Bikes. In *2024 14th International Conference on Electrical Engineering (ICEENG)* (pp. 142–144). [CrossRef]

[23] Yu, J., & others. (2024). Road Surface Defect Detection—From Image-Based to Non-Image-Based: A Survey. *IEEE Transactions on Intelligent Transportation Systems, 25*(9), 10581–10603. [CrossRef]

[24] Dilek, E., & Dener, M. (2023). Computer Vision Applications in Intelligent Transportation Systems: A Survey. *Sensors, 23*(6), 2938. [CrossRef]

[25] Safyari, Y., Mahdianpari, M., & Shiri, H. (2024). A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning. *Sensors, 24*(17), 5652. [CrossRef]

[26] Rathee, M., Bačić, B., & Doborjeh, M. (2023). Automated Road Defect and Anomaly Detection for Traffic Safety: A Systematic Review. *Sensors, 23*(12), 5656. [CrossRef]

[27] Ruggeri, A., Ficara, A., Sollazzo, G., Bosurgi, G., & Fazio, M. (2024, June). A Comparative Analysis of Deep Learning Approaches for Road Anomaly Detection. In *2024 IEEE Symposium on Computers and Communications (ISCC)* (pp. 1-6). IEEE. [CrossRef]

[28] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems, 28*.

[29] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Cham: Springer International Publishing. [CrossRef]

[30] Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017, July). Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 936-944). IEEE Computer Society. [CrossRef]

[31] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213-229). Cham: Springer International Publishing. [CrossRef]

[32] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

[33] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017, September). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 23-30). IEEE. [CrossRef]

[34] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T. Y., Cubuk, E. D., ... & Zoph, B. (2021, June). Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2917-2927). IEEE. [CrossRef]

[35] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139-144. [CrossRef]

[36] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017, October). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2242-2251). IEEE. [CrossRef]

[37] Karras, T., Laine, S., & Aila, T. (2019, June). A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4396-4405). IEEE. [CrossRef]

[38] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022, June). High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10674-10685). IEEE. [CrossRef]

[39] Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., ... & Katam, H. (2019). Blenderproc. *arXiv preprint arXiv:1911.01911*.

[40] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017, October). CARLA: An open urban driving simulator. In *Conference on robot learning* (pp. 1-16). PMLR.

[41] Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

[42] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

[43] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018, June). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2704-2713). IEEE. [CrossRef]

[44] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

[45] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018, June). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4510-4520). IEEE. [CrossRef]

[46] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

[47] Jocher, G., Chaurasia, A., Qiu, J., & Ultralytics. (n.d.). *YOLOv5: A family of object detection architectures and models* [Computer software]. GitHub. Retrieved February 8, 2025, from https://github.com/ultralytics/yolov5

[48] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023, June). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464-7475). IEEE. [CrossRef]

[49] Ultralytics. (n.d.). *YOLOv8* [Computer software]. GitHub. Retrieved February 8, 2025, from https://github.com/ultralytics/yolov8

[50] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., & Han, J. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems, 37*, 107984-108011.

[51] Ali, M. L., & Zhang, Z. (2024). The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers, 13*(12), 336. [CrossRef]

[52] Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2024, September). Yolov9: Learning what you want to learn using programmable gradient information. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland. [CrossRef]

[53] Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv preprint arXiv:2502.12524*.

[54] Ye, T., Qin, W., Zhao, Z., Gao, X., Deng, X., & Ouyang, Y. (2023). Real-time object detection network in UAV-vision based on CNN and transformer. *IEEE Transactions on Instrumentation and Measurement, 72*, 1-13. [CrossRef]

[55] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024, June). DETRs Beat YOLOs on Real-time Object Detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16965-16974). IEEE. [CrossRef]

[56] Lv, W., Zhao, Y., Chang, Q., Huang, K., Wang, G., & Liu, Y. (2024). Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv preprint arXiv:2407.17140*.

[57] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., ... & Darrell, T. (2020, June). BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2633-2642). IEEE. [CrossRef]

[58] Károly, A. I., Nádas, I., & Galambos, P. (2024). Synthetic Multimodal Video Benchmark (SMVB): Utilizing Blender for rich dataset generation. In *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (pp. 000065–000070). [CrossRef]

[59] Gangadharaiah, D. (2024). Synthetic data generation for enhanced computer vision applications: A CAD model and Blender approach. Institute for Photonic Systems, Ravensburg-Weingarten University. Available at https://hsbwgt.bsz-bw.de/frontdoor/deliver/index/docId/748/file/Gangadharaiah.pdf

[60] Basak, S., Javidnia, H., Khan, F., McDonnell, R., & Schukat, M. (2020, June). Methodology for building synthetic datasets with virtual humans. In *2020 31st Irish Signals and Systems Conference (ISSC)* (pp. 1-6). IEEE. [CrossRef]

[61] Károly, A. I., & Galambos, P. (2022, March). Automated dataset generation with Blender for deep learning-based object segmentation. In *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (pp. 000329-000334). IEEE. [CrossRef]

[62] Rahman, Z. U., Asaari, M. S. M., Ibrahim, H., Abidin, I. S. Z., & Ishak, M. K. (2024). Generative Adversarial Networks (GANs) for Image Augmentation in Farming: A Review. *IEEE Access, 12*, 179912–179943. [CrossRef]

[63] Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F.-Y. (2017). Generative Adversarial Networks: Introduction and Outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4), 588–598. [CrossRef]

[64] Raspberry Pi Ltd. (n.d.). Raspberry Pi 4 Model B [Product page]. Retrieved April 7, 2025, from https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[65] Raspberry Pi Ltd. (n.d.). Raspberry Pi 5 [Product page]. Retrieved April 7, 2025, from https://www.raspberrypi.com/products/raspberry-pi-5/

[66] NVIDIA Corporation. (n.d.). Jetson Nano [Product page]. NVIDIA Developer. Retrieved April 8, 2025, from https://developer.nvidia.com/embedded/jetson-nano

[67] Santos, R. C. C. D. M., Coelho, M., & Oliveira, R. (2024). Real-time object detection performance analysis using YOLOv7 on edge devices. *IEEE Latin America Transactions, 22*(10), 799-805. [CrossRef]

Tiago Silva received a Bachelor's degree in Software Engineering from the University of Trás-os-Montes and Alto Douro and completed a Master's degree in Computer Science at the University of Trás-os-Montes and Alto Douro (UTAD), Vila Real, Portugal, in 2025. (Email: tiagofsilva@utad.pt)

João Silva holds B.Sc. and M.Sc. degrees in Computer Science from the University of Trás-os-Montes and Alto Douro (UTAD), Portugal, where he is currently pursuing a Ph.D. in Informatics. His research focuses on Advanced Rider Assistance Systems (ARAS), with an emphasis on enhancing motorcycle safety through intelligent technologies. He has presented several papers at academic conferences and continues to expand his research contributions in this field. (Email: andres@utad.pt)

António Sousa completed his PhD in Engineering Sciences in 2011 at the University of Trás-os-Montes and Alto Douro (UTAD), where he is currently an Assistant Professor. He is a researcher at the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC) in the area of Computer Vision and Artificial Intelligence, having published more than 20 articles in specialized journals in these areas. He has participated as a researcher in several projects with contributions in both research areas. (Email: amrs@utad.pt)

Vítor Filipe is an Associate Professor with Habilitation in Electrical and Computing Engineering, specialising in Digital Signal Processing and Biomedical Image at the University of Trás-os-Montes e Alto Douro (UTAD), Portugal. He earned his bachelor's degree in Electrical and Computer Engineering in 1994 and his master's degree in Informatics from the University of Minho in 1998. He also holds a PhD in Electrical and Computer Engineering from UTAD, obtained in 2003. He is a Senior Researcher at the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC). His research focuses on medical image analysis, computer vision, and machine learning. Filipe has co-authored more than 160 Scopus-indexed articles, including over 60 in journals. He participated as a researcher in more than 25 projects, with financial support from scientific agencies and industrial partners. He was President of the School of Science and Technology at UTAD (2022-2025), which had over 2,000 students, in the period. (Email: vfilipe@utad.pt)