

Article

Offline UAV Inspection with Audited LLM-Assisted Analytics: A Deployable Framework Integrating Computer Vision and Natural Language Querying

Matias Soto ¹, Ricardo Vergara ¹, Pablo Ormeño-Arriagada ^{2,*} and Jorge Vasquez ¹

¹ Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Peñalolén, Santiago 7941169, Chile; matias.soto@edu.uai.cl (M.S.); ricardo.vergara@alu.ucm.cl (R.V.); jorge.vasquez.a@uai.cl (J.V.)

² Facultad de Ingeniería, Negocios y Ciencias Agroambientales, Universidad de Viña del Mar, Viña del Mar 2580022, Chile

* Correspondence: pablo.ormeno@uvm.cl

Abstract

Field-based infrastructure inspection often occurs under intermittent or absence connectivity, limiting the applicability of cloud-dependent analytical systems. Existing approaches primarily focus on detection accuracy or edge processing, but rarely address the integration of reliable analytics, data traceability, and deployment constraints within a unified framework. To address this gap, we present OffInspect-LLM, an offline inspection platform integrating controlled LLM-assisted analytical interaction. The system combines an inspection pipeline for object detection with a constrained natural language interface that generates validated SQL queries, ensuring safe and traceable data interaction. Experimental evaluation on a dataset of 1600 images demonstrated stable multi-seed held-out test performance, achieving a deployment-oriented mean held-out test mAP@50:95 of 0.617 across five random seeds, while the highest exploratory single-run validation result reached 0.714 under fixed initialization conditions. The primary deployment-oriented evaluation corresponds to the multi-seed held-out test performance rather than the peak single-run validation result. In addition to predictive performance, the system achieves per-image inference times below 3 s on GPU, reliable batch processing, and scalable geospatial visualization exceeding 10,000 detections. The primary contribution lies in the integration of detection, structured data management, and audited querying within an offline-first architecture, enabling traceable and deployment-oriented inspection workflows through constrained analytical interaction under realistic operational conditions.

Keywords: UAV inspection; computer vision; large language models (LLMs); text-to-SQL; offline systems



Academic Editor: João Paulo Carvalho

Received: 3 April 2026

Revised: 30 June 2026

Accepted: 14 July 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

1.1. Motivation and Problem Context

Field infrastructure inspections increasingly rely on Unmanned Aerial Vehicle (UAV) imagery to monitor distributed assets; however, high data volumes, intermittent connectivity, and limited edge analytics constrain reliable near-source processing [1,2]. Although DL detectors achieve strong accuracy in cloud-based environments, most frameworks assume stable connectivity and prioritize model-level performance over deployable system integration [3]. Consequently, a gap persists between detection accuracy and operational feasibility in remote settings. Moreover, UAV imagery captured under variable illumination, motion

blur, and sensor noise requires robust preprocessing to ensure stable inference [4]. Therefore, this work is motivated by the need to design an integrated inspection architecture that combines signal enhancement, statistically grounded detection stability, and secure offline analytics for autonomous field deployment.

1.2. Positioning Within Signal Processing

From a signal processing perspective, UAV inspection imagery poses a challenging enhancement and feature preservation problem due to non-uniform illumination, motion blur, occlusion, and sensor noise [5]. Therefore, improving defect visibility while preserving structural fidelity directly impacts detection performance. Classical methods such as histogram equalization and CLAHE enhance contrast while controlling noise amplification, whereas filtering and sharpening balance noise reduction with edge preservation. Importantly, recent studies confirm that contrast and edge enhancement significantly influence downstream vision tasks [6,7]. However, their systematic validation within offline UAV inspection pipelines remains limited. Consequently, this work is driven by the need to evaluate enhancement strategies under realistic deployment constraints, supporting both robustness and operational reliability [8].

1.3. Research Gap and Objective

While prior studies report strong detection accuracy using deep learning architectures, most assume cloud-connected pipelines or emphasize model-level performance without addressing signal enhancement robustness or deployment constraints. In particular, the interaction between preprocessing transformations and detector stability under stochastic training remains insufficiently quantified [9]. Moreover, limited attention has been given to embedding validated models within secure, auditable, and fully offline inspection systems suitable for field conditions. Auditability refers to operational traceability rather than formal cryptographic verification. Consequently, this gap motivates a more integrated perspective that considers not only detection performance but also robustness, reproducibility, and operational reliability under realistic deployment scenarios.

To address these gaps, this study investigates the following research questions:

1. RQ1: How do different signal enhancement strategies influence UAV-based defect detection performance and model stability under stochastic training variability?
2. RQ2: Is it feasible to integrate a statistically validated UAV-based defect detection model within a fully offline inspection architecture that enables audited, read-only natural language querying while maintaining acceptable operational performance?

In this work, audited interaction refers to traceable read-only query execution supported by structured validation, query logging, controlled schema access, and transparent result exposure. Furthermore, although operational feasibility is quantitatively evaluated through system-level metrics, the transparency of the LLM-assisted component is addressed through constrained query execution and operational auditability rather than formal user-centered evaluation. Accordingly, the framework prioritizes controlled analytical interaction and auditable system behavior within practical inspection workflows.

1.4. Proposed Solution and Contribution

To address this gap, we introduce OffInspect-LLM, an offline-first inspection platform integrating controlled LLM-assisted analytical interaction. The system combines UAV-based defect detection, georeferenced inspection management, and constrained text-to-SQL querying within a desktop environment built using FastAPI, SQLite, Huey, and an Electron, React frontend. Specifically, the framework supports structured ingestion, tiling, post-processing, severity annotation, and cost attribution while prioritizing near-source analysis

to reduce cloud dependency and preserve operational transparency. In addition, constrained LLM interaction is implemented through controlled database views and restricted SQL constructs to support traceable and secure analytical workflows [10–12]. Evaluation further incorporates mAP@50:95, latency, and reliability metrics to align predictive performance with deployment-oriented feasibility [13]. Importantly, the framework focuses on offline post-acquisition inspection analytics rather than autonomous UAV navigation or real-time flight control, supporting secure and interpretable decision support under realistic operational constraints. The present work assumes non-privileged analytical users operating under schema-restricted query environments and does not evaluate adversarial prompt injection under hostile threat conditions.

1.5. Structure of the Paper

The remainder of this paper is organized as follows. Section 2 reviews related work in UAV-based inspection, signal enhancement, and LLM-assisted querying. Section 3 details the system architecture, signal processing pipeline and presents the experimental setup and robustness evaluation. Section 4 reports quantitative and qualitative results. Section 5 discusses system implications, security considerations, and limitations. Finally, Section 6 concludes the study and outlines future research directions.

2. Related Work

The existing literature relevant to this work spans three interconnected domains: (i) signal processing and deep learning for UAV-based defect detection, (ii) system-level architectures for infrastructure inspection under connectivity constraints, and (iii) natural language interfaces for structured data querying. The following subsections reorganize prior contributions along these thematic axes to clarify positioning and highlight unresolved gaps addressed in this study.

2.1. Signal Processing and UAV-Based Defect Detection

Although DL has significantly improved UAV defect detection through models such as YOLO and MASK R-CNN, achieving strong performance in aerial inspection tasks, most operational pipelines still rely on centralized or cloud-based processing, limiting reliability in environments with intermittent connectivity [14]. From a signal processing perspective, robust detection further requires extracting discriminative structural patterns under non-uniform illumination, scale variation, motion blur, and occlusion [15]. Consequently, performance depends not only on model architecture but also on preprocessing transformations affecting luminance, spatial frequency, and contrast. However, the interaction between classical enhancement techniques and modern detectors remains insufficiently quantified, particularly under deployment constraints [16]. Therefore, there is a clear need for an integrated perspective that jointly considers model performance, preprocessing strategies, and system-level robustness to enable reliable and field-ready UAV inspection systems.

2.2. Offline and Edge-Based Inspection Architectures

While offline-first and edge computing architectures improve operational resilience by reducing dependence on external infrastructure, existing systems often prioritize efficiency over structured analytics and interactive exploration [17]. In parallel, UAV inspection studies typically report results under fixed configurations, overlooking the impact of preprocessing strategies and stochastic variability. From a signal processing perspective, transformations such as contrast normalization, smoothing, and sharpening can significantly influence detection stability across random initializations. However, these interactions remain insufficiently explored under deployment constraints [18]. Therefore, this work is motivated by the need to systematically analyze how preprocessing affects robustness,

while also integrating computer vision pipelines with embedded databases and local task orchestration to enable reliable, autonomous, and field-ready inspection systems [19].

2.3. LLM-Based Text-to-SQL Systems for Industrial Analytics

It is important to distinguish between model-level performance, measured by mAP, precision, and recall, and system-level performance, which includes latency, scalability, and user interaction. However, prior work often treats these dimensions independently, limiting real-world applicability [20]. In parallel, LLMs have advanced text-to-sql generation, enabling flexible access to structured data through benchmarks such as Spider and constrained inference strategies. Nevertheless, challenges related to security, auditability, and controlled execution persist in mission-critical contexts. Consequently, mitigation strategies such as restricted views and read-only enforcement have been proposed, yet their integration into inspection systems remains limited. Therefore, this study adopts an integrated approach, combining both performance layers within a unified offline architecture to enable secure, reproducible, and reliable LLM-driven analytics [21,22].

2.4. Contributions of This Work

This study advances the state of the art by proposing a unified and deployment-oriented framework for UAV-based inspection. First, it introduces a signal-processing-aware evaluation, systematically analyzing preprocessing strategies such as contrast normalization, smoothing, sharpening, and white balance under realistic acquisition conditions. Second, it incorporates statistical robustness through multi-seed training, reporting mean, standard deviation, and 95% confidence intervals to ensure reproducibility. Third, it develops a fully offline inspection platform integrating YOLOv8n-based detection [23], SQLite persistence [24], Huey task orchestration, and geospatial visualization. In addition, it enables audited LLM-based querying through controlled text-to-SQL mechanisms, ensuring secure and traceable interaction. Finally, it establishes a clear linkage between model-level metrics and system-level performance, aligning algorithmic accuracy with operational deployment constraints.

2.5. Identified Research Gap

The reviewed literature demonstrates substantial progress in deep learning based defect detection, resilient offline and edge computing architectures [25], and natural language interfaces for structured querying. However, these domains remain largely disconnected in practical deployment settings. Although UAV inspection detectors often achieve strong predictive performance [26], the influence of signal enhancement, stochastic variability, and multi-seed robustness evaluation is rarely examined, with many studies relying on isolated single-run results. Similarly, while offline-first architectures improve operational resilience, few systems integrate statistically validated vision models within secure and auditable workflows. LLM-based text-to-SQL methods also enable flexible querying [12], yet their controlled deployment in safety-critical offline environments remains limited. Therefore, the principal contribution of this work is the integration of statistically validated UAV inspection, offline embedded persistence, geospatial processing, and constrained LLM-assisted querying within a unified deployment-oriented analytical ecosystem. To further substantiate the identified research gap, Table 1 compares the proposed framework with the closest related integrated inspection systems across the principal architectural characteristics emphasized in this work.

Table 1. Comparison between representative integrated inspection systems and the proposed framework.

System	UAV Inspection	Offline Execution	Statistical Robustness Evaluation	Constrained LLM Querying
Lekidis et al. [10]	Yes	Yes	No	No
Husom et al. [13]	No	Yes	No	No
Proposed framework	Yes	Yes	Yes	Yes

3. Materials and Methods

This section describes the methodological framework underlying the proposed offline inspection system, integrating computer vision, signal processing, and LLM-based analytics within a unified architecture. First, the system design is presented, detailing the offline-first architecture and its core components, including data management, task orchestration, and user interaction. Next, the vision processing pipeline is introduced, covering data ingestion, preprocessing strategies, and detection model configuration. Subsequently, the experimental setup is defined, including dataset characteristics, training protocols, and evaluation metrics used to assess both detection performance and system-level behavior. In parallel, the audited LLM-based querying mechanism is described, emphasizing controlled text-to-SQL generation and security constraints. Finally, the section outlines the robustness evaluation methodology, incorporating multi-seed analysis and statistical reporting to ensure reproducibility and reliability under realistic deployment conditions. The proposed platform processes real UAV-acquired inspection imagery within an operational offline analytical workflow, including ingestion, georeferencing, defect detection, persistence, and controlled analytical interaction. However, autonomous UAV flight experimentation, onboard real-time inference, and direct flight-control integration remain outside the scope of the present study.

3.1. System Architecture Overview

The platform follows a modular offline-first design, combining local persistence, asynchronous job queues, and a desktop-based interactive interface. Specifically, it integrates FastAPI [27], SQLite, HUEY, and an ELECTRON-based frontend to enable autonomous operation without Internet connectivity. This design supports resilient deployment through local data management, asynchronous processing, and packaged inference models. Figure 1 summarizes the system architecture. At the backend, REST endpoints manage core entities and orchestrate processing tasks, while HUEY schedules tiling, inference, and post-processing with controlled execution policies. SQLITE ensures efficient persistence through WAL mode, indexing, and backup strategies. At the frontend, the ELECTRON + REACT interface provides integrated modules for monitoring, geospatial visualization, task tracking, and assisted interaction, reflecting a deliberate effort to align system usability with deployment constraints.

The database schema is designed to support the complete lifecycle of UAV-based inspection workflows, from mission definition to defect analysis. At its core, the structure organizes data into five interconnected entities: users, inspections, flights, images, and detections. Each inspection aggregates multiple flights, capturing contextual and environmental metadata such as weather conditions, altitude, and execution timestamps. Flights generate collections of images, which are processed and linked to detection outputs containing spatial (e.g., bounding boxes, coordinates) and semantic information (e.g., severity, category, confidence scores). The schema ensures traceability across all stages of the pipeline, enabling efficient querying, auditability, and scalable storage of both raw and processed inspection data. As shown in Figure 2, the proposed relational model structures

the UAV inspection workflow through interconnected entities that ensure traceability from data acquisition to defect detection.

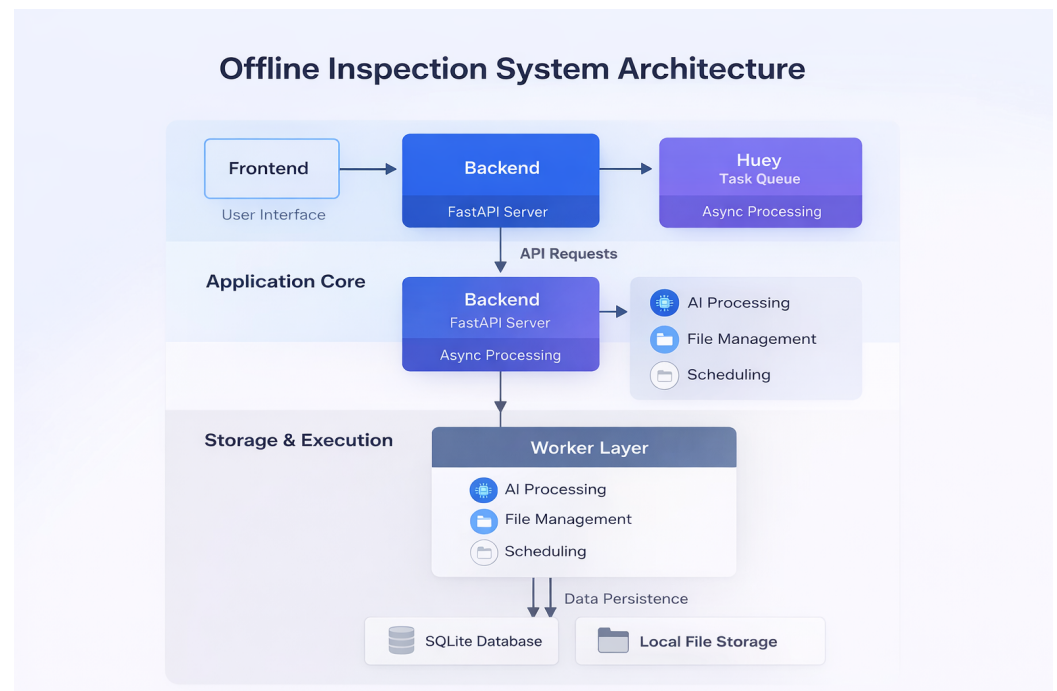


Figure 1. Offline-first inspection system architecture. The platform integrates a desktop-based frontend with a FastAPI backend and a Huey task queue to enable asynchronous processing under limited connectivity. A dedicated worker layer executes core tasks, including AI-based defect detection, file management, and scheduling, while ensuring traceable operations. Processed data and metadata are persistently stored using an embedded SQLite database and local file storage. This design supports scalable ingestion, georeferenced analysis, and secure, auditable workflows, enabling reliable UAV-based inspection in distributed and connectivity-constrained environments.

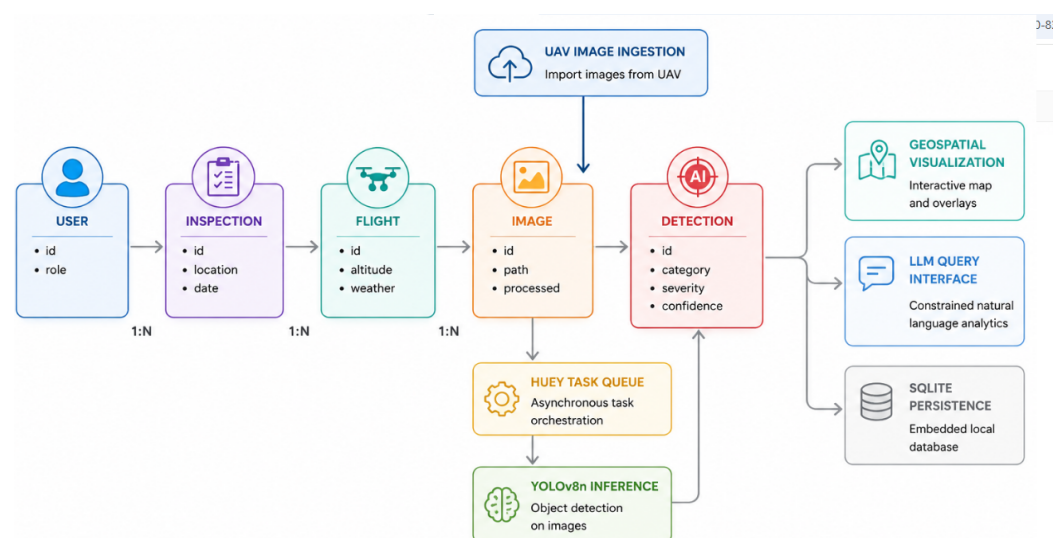


Figure 2. Operational architecture and data interaction workflow of the proposed offline-first inspection platform. The figure illustrates the interaction between UAV image ingestion, embedded SQLite persistence, asynchronous task orchestration through Huey, YOLOv8n-based computer vision inference, geospatial visualization, and constrained LLM-assisted analytical querying across the integrated inspection workflow. The relational structure between users, inspections, flights, images, and detections is additionally represented to contextualize the underlying data flow within the operational pipeline.

The architecture prioritizes constrained analytical interaction and traceable execution rather than unrestricted autonomous query generation. Figure 2 presents a high-level operational abstraction of the relational workflow rather than a complete engineering-level database specification. Specifically, the underlying SQLite schema incorporates indexed primary and foreign key relationships linking users, inspections, flights, images, and detections to preserve traceability across ingestion, inference, and analytical interaction stages. Accordingly, the database design supports accountable data persistence and controlled analytical access within the offline inspection framework. Complete implementation-level schema details will be provided in the public-repository-accompanying publication to support reproducibility and transparent methodological validation.

3.2. Vision Processing Pipeline

The proposed vision pipeline is based exclusively on object detection using YOLOv8n and does not incorporate semantic or instance segmentation mechanisms within the evaluated framework. The vision pipeline performs ingestion, georeferencing, optional tiling, batch inference, and post-processing to generate actionable inspection artifacts. Specifically, UAV imagery [28] is processed through a structured workflow that balances robustness and efficiency using a four-class detection model with domain-specific preprocessing. The pipeline includes data validation, geotagging via Exchangeable Image File Format (EXIF) or Comma-Separated Values (CSVs) with Ground Sampling Distance (GSD) computation, and resolution-aware tiling with configurable parameters. Subsequently, batch inference is performed using class-specific thresholds and Non-Maximum Suppression (NMS), followed by post-processing for severity and cost estimation, producing outputs in GEOJSON or CSV formats [29]. In parallel, the experimental workflow spans acquisition, annotation, preprocessing, training, and statistical robustness evaluation, culminating in system integration. Figure 3 summarizes this end-to-end process, reflecting a deliberate effort to connect model development with deployable inspection workflows.

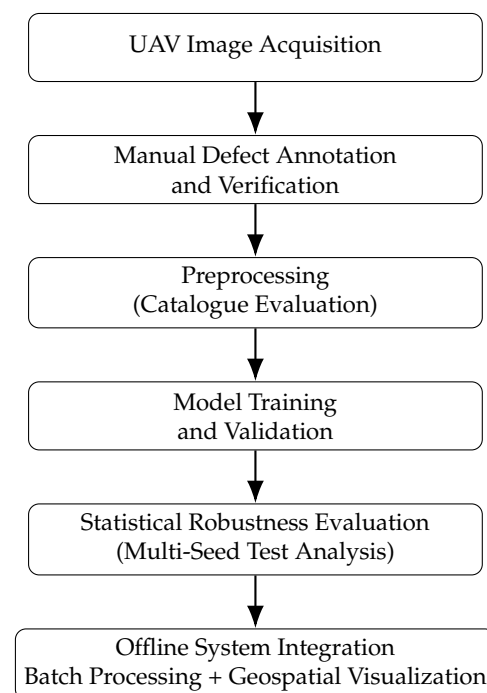


Figure 3. End-to-end experimental workflow from UAV acquisition to offline deployment and geospatial inspection integration.

Severity levels and associated operational costs are not predicted by the detection model but are manually configured by domain experts or operational users through pre-defined inspection categories after defect detection. This design enables downstream quantitative analysis and operational prioritization without introducing additional predictive uncertainty into the computer vision pipeline.

3.3. Preprocessing Strategy and Robustness Evaluation

To evaluate the sensitivity of the detection model to variations in illumination, noise, contrast, and structure, a controlled catalogue of preprocessing pipelines was defined. Rather than solely improving performance, the aim is to understand how visual transformations influence robustness in field-acquired UAV imagery. The catalogue includes baseline, contrast enhancement using CE and Contrast Limited Adaptive Histogram Equalization (CLAHE), smoothing, combined pipelines, color correction, and edge-based representations. From a signal processing perspective, these transformations alter luminance distribution, spatial frequency, and contrast statistics, directly impacting feature extraction. In particular, CLAHE enhances local contrast, smoothing reduces noise, and sharpening reinforces contours. All configurations were evaluated under identical conditions to ensure comparability and support a principled sensitivity analysis [16]. Table 2 summarizes the evaluated strategies.

Table 2. Image preprocessing configurations evaluated.

Category	Name	Operations
Baseline	Normal	None
Contrast Enhancement	CE CLAHE	Global histogram equalization Local contrast equalization
Smoothing	Blur GaussianBlur	Median blur (3×3) Gaussian blur (5×5)
Combined Enhancements	GaussianBlurContrast CLAHE_GB CLAHE_sharpening CLAHE_GB_CE_sharpening	Gaussian blur $\rightarrow$ CE CLAHE $\rightarrow$ Gaussian blur CLAHE $\rightarrow$ sharpen CLAHE $\rightarrow$ Gaussian blur $\rightarrow$ CE $\rightarrow$ sharpen
Color Correction	WhiteBalance	Gray-world white balance
Structural Emphasis	Edge	Canny edge map (3-channel)

Preprocessing configurations were initially screened using validation performance under identical training conditions, after which the selected configuration underwent an independent multi-seed held-out test evaluation to assess robustness and generalization.

3.4. Vision Model Training

A four-class object detection model was trained using domain-specific preprocessing to balance accuracy, robustness, and computational efficiency. Specifically, the dataset includes 1600 images across the classes Broken, Flashover Damage, No Issues, and String. The dataset includes multiple operational inspection contexts captured under varying illumination, background composition, and viewing perspectives, although broader diversity across weather conditions, sensor platforms, and infrastructure types remains limited. However, the evaluated acquisition conditions remain relatively constrained, and therefore, the reported results should not be interpreted as fully representative of broader operational environments involving substantially different weather conditions, sensor platforms, infrastructure configurations, or geographic contexts. Training was conducted for 100 epochs

with a dataset partition consisting of 70% training, 15% validation, and 15% held-out testing samples, a batch size of 8, and preprocessing using CLAHE and sharpening. The multi-seed robustness evaluation was conducted using the CLAHE + sharpening configuration selected during exploratory validation analysis. Although augmentation increased the effective training volume, the underlying number of unique acquisition scenarios remained limited, and therefore, the dataset should be interpreted as a constrained experimental benchmark rather than a comprehensive representation of real-world inspection variability. Dataset partitioning was performed at the base-image level prior to augmentation to prevent information leakage between training, validation, and test subsets. However, to mitigate dependence on a single stochastic initialization, additional robustness experiments were performed. Importantly, the dataset was assembled from Roboflow subsets (<https://universe.roboflow.com/claudio-tapia-lb7t2/bad-insulator-detector/dataset/3>, accessed 30 June 2026), with approximately 400 unique base images. To prevent data leakage, splitting was applied at the base-image level prior to augmentation. Subsequently, a 70/15/15 train, validation, test partition and simple augmentations were used to support generalization. Figure 4 illustrates representative samples of the four target classes used in the dataset, highlighting the visual characteristics and annotation scheme adopted for model training. Specifically, each class is color-coded to facilitate interpretability, allowing clear differentiation between normal conditions and defect types. This visualization provides insight into the variability of UAV-acquired imagery and supports understanding of the classification task addressed by the detection model. The four detection categories were treated as mutually exclusive classes during annotation. Specifically, Broken and Flashover Damage correspond to defect conditions, whereas String and No Issues represent structurally valid operational components used as contextual reference categories within the inspection workflow. Ambiguous samples were manually reviewed to ensure consistent label assignment across operational conditions and to reduce semantic overlap between defect and non-defect categories. The No Issues category corresponds to structurally valid components intentionally annotated to distinguish defect-free operational conditions from damaged or degraded instances captured under similar acquisition contexts. To improve interpretability and clarify the semantic role of each detection category, Table 3 summarizes the ontology adopted during annotation together with the distribution of each detection category. Specifically, the dataset includes both defect-related classes and operational reference structures, enabling the detection model to distinguish damaged components from structurally valid conditions within realistic UAV inspection scenarios. The reported distribution also highlights the inherent class imbalance of the dataset, with the No Issues category representing the majority of annotated objects, reflecting the predominance of defect-free components typically encountered during routine UAV inspections.

Table 3. Semantic interpretation and distribution of the detection classes used in the UAV inspection dataset.

Class	Semantic Role	Distribution	Description
Broken	Defect	(6%)	Structural damage affecting the physical integrity of the inspected component.
Flashover Damage	Defect	(14%)	Electrical degradation associated with flashover-related surface or insulation damage.
String	Reference Structure	(10%)	Operational infrastructure component included to provide contextual structural information during inspection.
No Issues	Normal Condition	(69%)	Structurally valid component intentionally annotated to represent defect-free operational conditions.

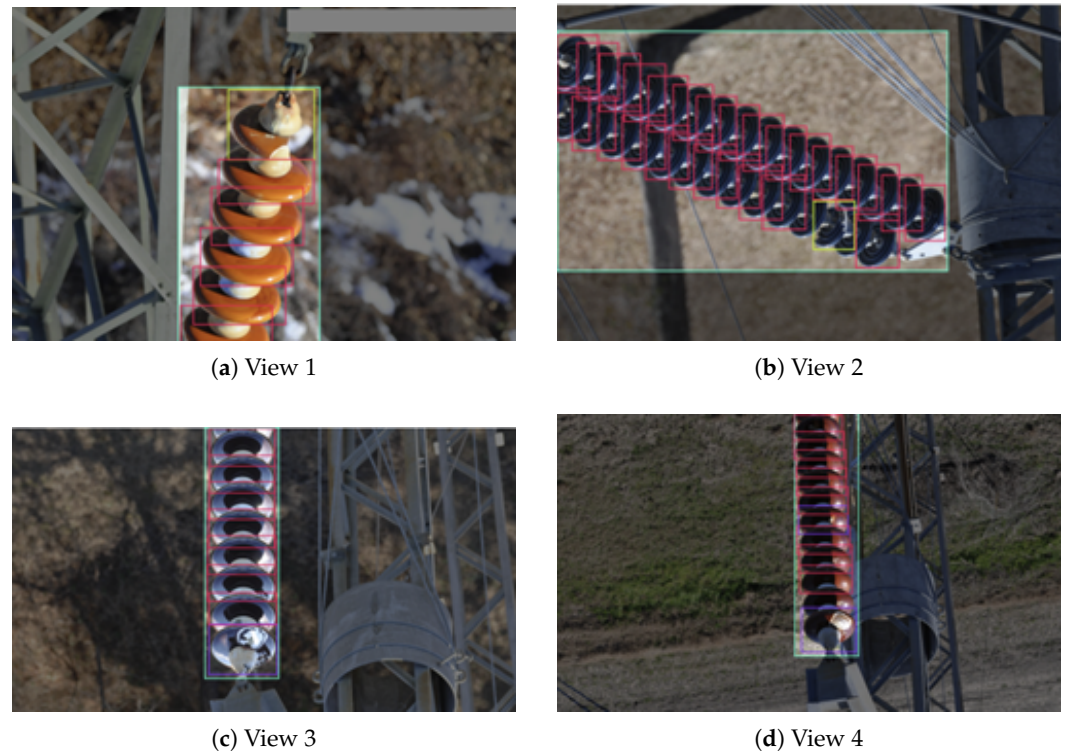


Figure 4. Representative examples of the four detection categories used during model training. Bounding-box annotations identify the classes, No Issues (cyan), Broken (red), Flashover Damage (yellow), and String (purple) across different UAV inspection scenarios. The color coding is used exclusively for visual differentiation of the annotated categories and does not represent confidence scores or severity levels.

The detection backbone is YOLOv8n (Ultralytics), selected for its balance between computational efficiency and detection accuracy in edge and offline contexts. Training was conducted for 100 epochs with a batch size of 8, maintaining default anchor-free settings. To ensure reliability, stochastic variability was assessed by retraining the selected preprocessing configuration across five random seeds (11, 22, 33, 44, 55) under identical conditions. Performance was evaluated on the test set using mean, standard deviation, and 95% confidence intervals for mAP@50:95, mAP@50, precision, and recall, yielding an average mAP@50:95 of approximately 0.617. Methodologically, development followed two stages—preprocessing comparison and multi-seed robustness evaluation—reflecting a deliberate effort to ensure reproducibility prior to deployment. Training was performed using the YOLOv8n default optimization configuration with stochastic gradient descent (SGD), an initial learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005. A cosine learning-rate schedule was employed during training. In addition to preprocessing transformations, conservative YOLOv8 augmentation settings were employed, including limited scaling, horizontal flipping, color-space perturbation, and translation, while aggressive strategies such as MixUp remained disabled. Training was performed using an input image resolution of 640×640 pixels. Model selection was based on the best validation mAP@50:95 checkpoint (best.pt), which was subsequently used for held-out test evaluation. Early stopping was enabled using a patience threshold of 15 epochs without validation improvement. Training was conducted using the Ultralytics YOLOv8 implementation under deterministic execution settings to improve experimental reproducibility. All experiments were conducted using the Ultralytics YOLOv8 framework and associated default optimization settings available at the time of experimentation.

3.5. Audited LLM Assistant with SQL Query Workflow

To enable accessible yet secure interaction with inspection data, the proposed multi-agent LLM workflow translates natural language into read-only SQL queries under strict validation constraints. The process begins with LLM0, which classifies user input as either a general or data-oriented request using schema-level awareness. Subsequently, generated queries are validated before execution to reduce the likelihood of unintended or destructive operations and ensure controlled data exposure. In addition, the framework returns structured outputs with textual explanations and optional visualizations, thereby improving usability for non-technical users. While recent text-to-SQL approaches demonstrate strong semantic parsing capabilities, security and auditability remain essential in real-world environments. Therefore, the proposed design reflects a deliberate effort to balance usability, transparency, and operational control within a reliable analytical framework [21,30]. The validation stage further enforces read-only execution through restricted SQL constructs, schema-level constraints, controlled database views, and predefined query verification rules.

If a query is classified as general, it is routed to LLM1, which provides guidance, clarifies system constraints, and reformulates requests into structured analytical queries without accessing the database. Conversely, data-oriented requests are forwarded to LLM2, which generates SQL statements from the relational schema. These statements are subsequently validated by LLM3, responsible for enforcing read-only execution and rejecting potentially destructive operations according to predefined validation rules. If approved, LLM2 executes the query using batch-based processing to maintain operational stability. Subsequently, LLM4 analyzes the retrieved data and suggests complementary visualizations. This modular architecture reflects a deliberate effort to balance usability, transparency, and security within practical analytical workflows. Importantly, the multi-agent structure represents logical orchestration stages rather than independently deployed foundation models and remains compatible with both local and abstracted LLM backends. Detailed memory consumption and disk footprint profiling were not systematically evaluated and therefore remain relevant directions for future deployment-oriented optimization.

To ensure transparent interaction with inspection data, each query processed by the multi-agent LLM workflow returns a natural language explanation, a structured data table reflecting retrieved information, and optional visualizations when supported by the data. In this way, the system enhances interpretability while maintaining clarity for non-technical users. Figure 5 illustrates the coordinated sequence of agents responsible for query classification, SQL generation, validation, execution, and result interpretation. Importantly, this design enforces controlled execution and traceability, ensuring that all interactions remain auditable. Therefore, the workflow reflects a deliberate effort to integrate usability, transparency, and security within a structured analytical framework suitable for real-world deployment.

The proposed auditability mechanism combines query logging, controlled execution constraints, schema-level validation, and explicit exposure of retrieved outputs to support observable analytical interaction. The proposed orchestration framework is intentionally model-agnostic and compatible with different locally hosted or externally abstracted language model backends, depending on deployment constraints and available computational resources. To prioritize deterministic and auditable execution, inference was configured using constrained low-temperature generation settings and schema-restricted prompting strategies rather than unrestricted conversational interaction. The orchestration layer was intentionally designed to remain model-agnostic and compatible with interchangeable local or externally abstracted LLM backends, depending on deployment constraints and available computational resources.

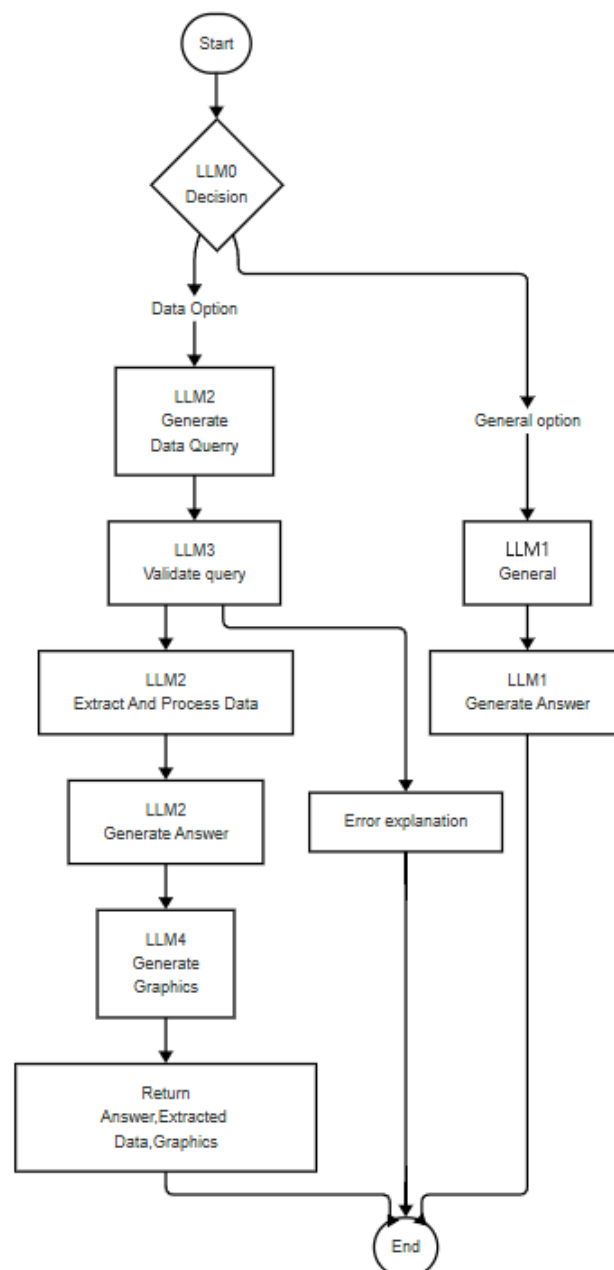


Figure 5. Multi-agent LLM workflow for secure text-to-SQL interaction. The architecture includes query classification (LLM0), user guidance (LLM1), SQL generation (LLM2), safety validation (LLM3), and result interpretation with optional visualization (LLM4). The design enforces read-only execution, structured outputs, and controlled data access within an auditable analytical pipeline.

3.6. Evaluation Scope of the LLM Assistant

The evaluation of the LLM component focuses on system-level feasibility rather than formal correctness benchmarking. In natural language to SQL translation, no unique ground truth query exists because multiple semantically equivalent SQL formulations may generate identical valid outputs. Consequently, direct comparison against predefined expected queries is not a reliable evaluation strategy. Instead, the present study prioritizes controlled execution, structural consistency, and safe database interaction over benchmark-driven accuracy assessment. Operational indicators such as hallucination frequency, invalid schema generation, adversarial query handling, and execution latency, therefore, remained outside the scope of the feasibility-oriented evaluation. Moreover, no formal user-centered assessment was conducted, and usability, interpretability, and interaction quality should

not yet be considered empirically validated dimensions of the conversational workflow. Accordingly, the LLM subsystem should be interpreted as a controlled proof of concept integration layer intended to demonstrate accountable execution and constrained analytical interaction within an offline inspection framework. Accordingly, semantic equivalence between retrieved outputs rather than exact SQL string matching represents the more meaningful evaluation target for deployment-oriented analytical interaction.

3.7. Experimental Setup

Experiments were conducted on a GPU-enabled workstation to evaluate both detection accuracy and processing latency, with additional CPU-only execution to assess performance degradation under non-accelerated conditions. This setup enables a more realistic understanding of system behavior across deployment scenarios. Detector performance was measured using mAP@50:95, ensuring consistency with standard evaluation protocols. In parallel, system-level performance was assessed through per-image processing time, covering tiling, inference, and post-processing stages, along with job completion rate. Furthermore, user interface responsiveness was evaluated using indicators such as layer loading time, clustering efficiency, and pan and zoom interaction. This combined evaluation reflects a deliberate effort to align model-level accuracy with operational performance in practical inspection environments. Table 4 summarizes the evaluation metrics used to assess detection performance, providing standardized measures of accuracy, localization quality, and the impact of preprocessing strategies.

Table 4. Evaluation metrics used for detection performance analysis.

Metric	Description
mAP@{50:95}	Mean Average Precision computed over IoU thresholds from 0.50 to 0.95 in increments of 0.05, providing a comprehensive measure of detection performance across varying localization strictness.
Δ vs. Normal	Difference in mAP@{50:95} relative to the baseline configuration (normal), used to quantify the impact of preprocessing strategies on detection performance.
mAP@50	Mean Average Precision at a fixed IoU threshold of 0.50, reflecting detection performance under a less strict localization criterion.
Precision	Ratio of true positive detections to all predicted positives, indicating the model's ability to avoid false positives.
Recall	Ratio of true positive detections to all actual positives, measuring the model's ability to detect all relevant objects.

3.8. Reproducibility and Code Availability

To ensure transparency and reproducibility, the complete experimental pipeline is maintained in a version-controlled GitHub repository (version 1.0) repository (The repository link will be provided upon publication to preserve double blind review integrity). The repository contains all components required for replication, including preprocessing routines, configuration files, deployment scripts, and reproducibility instructions. However, to preserve the integrity of the double blind review process, the public repository URL is intentionally omitted during peer review. Accordingly, the full implementation will be publicly released upon publication to support transparent validation, methodological traceability, and reproducible future research.

4. Results

This section presents the quantitative and qualitative evaluation of the proposed system. First, detection performance across preprocessing configurations is analyzed. Next, statistical robustness is assessed through multi-seed experiments. Finally, system-level performance is reported, linking model accuracy with operational efficiency to validate the framework under realistic UAV inspection conditions.

4.1. Preprocessing Comparison

To evaluate the influence of signal enhancement on detection performance, all preprocessing configurations in Table 2 were trained under identical YOLOv8n conditions for 100 epochs. Initial screening was performed on the validation set to identify candidate configurations for robustness analysis. The highest single-run mAP@50:95 was obtained using CLAHE + sharpening and was subsequently incorporated into the comparative held-out test evaluation reported in Table 5. However, this result reflects fixed stochastic initialization conditions and should therefore be interpreted only as an exploratory upper-bound estimate rather than as stable deployment-oriented performance. Accordingly, the final evaluation was based on multi-seed held-out testing to assess consistency under unseen data conditions and reduce sensitivity to stochastic variability. Overall, the model demonstrated competitive multi-class detection capability under realistic UAV acquisition conditions, supporting the feasibility of integrating automated defect localization within an offline inspection framework.

Table 5. Impact of preprocessing strategies on held-out test-set detection performance. The table includes both exploratory preprocessing configurations and the CLAHE + sharpening pipeline selected during validation-based screening prior to multi-seed robustness evaluation.

Preprocessing	mAP@50:95	Δ vs. Normal	mAP@50	Precision	Recall
Normal	0.591	0.000	0.740	0.670	0.779
White Balance	0.605	+0.014	0.753	0.719	0.763
Gaussian Blur	0.585	−0.006	0.729	0.748	0.721
Gaussian Blur + CE	0.587	−0.004	0.740	0.725	0.689
CLAHE	0.566	−0.025	0.714	0.650	0.725
CLAHE + GB	0.585	−0.006	0.721	0.704	0.712
CE	0.576	−0.015	0.728	0.748	0.710
CLAHE + Sharpening	0.570	−0.021	0.709	0.758	0.612
CLAHE + GB + CE + Sharpening	0.539	−0.052	0.688	0.744	0.658
Blur (Median)	0.561	−0.030	0.702	0.750	0.709
Edge Only	0.463	−0.128	0.602	0.708	0.581

4.2. Detection and System Performance Analysis

Table 6 summarizes the core validation metrics and operational performance indicators. The object detection model achieved a mAP@50:95 of 0.714 across 1600 images and four defect classes, supporting reliable localization under realistic UAV acquisition conditions. In addition, GPU-based tiled batch processing maintained inference latency below three seconds per image, enabling near-real-time field analysis. Operational stability was further supported by a job completion rate exceeding 95%, with most failure cases associated with interrupted ingestion processes rather than catastrophic model or database failures. Moreover, interactive map rendering remained stable during exploratory visualization tests involving more than 10,000 clustered spatial points, highlighting scalability for

large inspection campaigns. However, these operational metrics should be interpreted as implementation-focused feasibility observations rather than formal benchmarking results, since frame time profiling, memory consumption analysis, and constrained latency measurements were not systematically evaluated. Figure 6 qualitatively confirms the reported quantitative behavior.

Table 6. Summary of detection and system-level performance under realistic deployment conditions. The primary performance indicator corresponds to the multi-seed held-out test mAP@50:95, reported as the mean across five independent random initializations. The peak validation result is included for preprocessing-stage screening purposes only and should not be interpreted as the primary deployment-oriented evaluation metric. Detection performance was evaluated on 1600 UAV images across four defect classes, while system metrics reflect GPU-based tiled inference and large-scale interactive visualization tests.

Metric	Value	Evaluation Condition
Deployment mAP@50:95 (5-seed test mean)	0.617	Held-out test set, mean across five independent random initializations
Peak validation mAP@50:95	0.714	Best single-run validation result obtained during preprocessing-stage screening
Inference time per image	<3 s	GPU, tiled batch processing
Job completion rate	≥95%	Test dataset execution
Interactive map rendering (10k+ points)	Smooth	Clustering enabled

Figure 6 presents representative detection outputs generated by the trained model on real UAV inspection imagery. The predicted bounding boxes, annotated with confidence scores, illustrate the model's ability to distinguish between defect categories such as Broken components and No Issues instances within vertically structured infrastructure elements. The results demonstrate consistent localization performance despite variations in illumination, background vegetation, and perspective distortions. Confidence values indicate stable classification behavior across multiple adjacent components, supporting the quantitative performance reported in Table 6. Overall, the qualitative outputs confirm that the detection model generalizes effectively to realistic field conditions and maintains structural coherence in densely repeated object configurations.



Figure 6. Representative detection outputs on real UAV inspection images. Bounding boxes illustrate predicted defect instances across the four target classes. The examples demonstrate consistent localization performance under varying illumination, perspective, and background complexity.

4.3. Impact of Preprocessing on Detection Performance

The preprocessing selection stage should be interpreted as an exploratory validation-based screening procedure rather than a statistically conclusive ranking methodology. Table 5 reveals qualitative differences in how preprocessing strategies influence detection behavior, although several performance variations remained comparable to the observed stochastic variability. Consequently, these differences should not be interpreted as definitive evidence of preprocessing superiority without broader statistical validation. From a signal processing perspective, preprocessing transformations alter luminance distribution, spatial frequency characteristics, and edge preservation behavior, thereby influencing downstream feature extraction stability and detection consistency. Importantly, the CLAHE + sharpening configuration initially selected during exploratory validation was reevaluated on the held-out test set to assess consistency under real-world conditions. Overall, light preprocessing transformations produced the most competitive results, with White Balance suggesting improved feature consistency under variable illumination, whereas aggressive contrast enhancement and smoothing often introduced marginal or negative effects by distorting discriminative visual patterns.

Reported preprocessing results correspond to exploratory single-run evaluation and were used exclusively for preliminary configuration screening prior to multi-seed robustness assessment. Notably, configurations such as CE and Gaussian Blur increase precision while reducing recall, reflecting a trade-off where noise reduction improves confidence but limits detection sensitivity. Furthermore, aggressive pipelines combining multiple transformations lead to performance degradation, highlighting the risk of cumulative signal distortion. The edge-only configuration shows the most pronounced decline, confirming that texture and color information remain essential for reliable detection. Collectively, these findings suggest that balanced, minimally invasive preprocessing strategies are preferable for maintaining detection robustness in UAV imagery. Figure 7 illustrates the impact of preprocessing strategies on detection performance, highlighting variations in mAP@50:95 across configurations, which differ from validation-based selection due to stochastic variability and generalization effects.

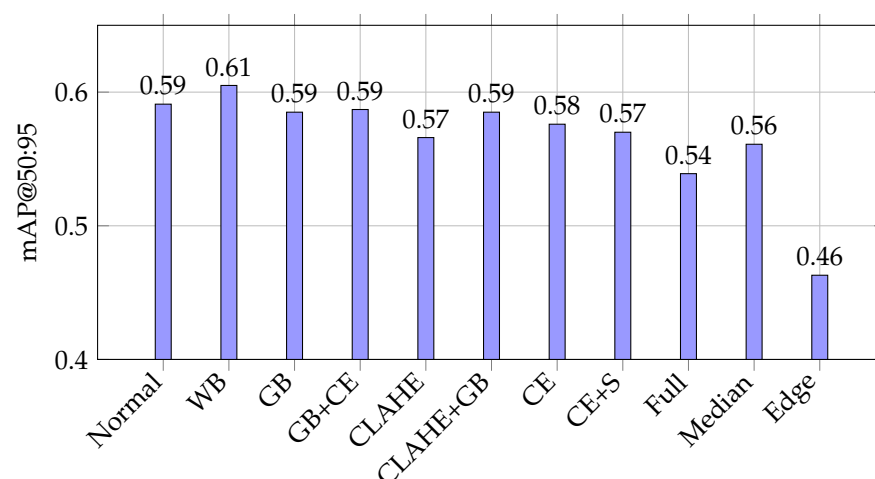


Figure 7. Comparison of mAP@50:95 across preprocessing strategies evaluated under identical held-out test conditions.

Although certain preprocessing configurations exhibited slightly higher single-run performance, these differences remained comparable to the observed stochastic variability and therefore should be interpreted cautiously.

4.4. Training Robustness Across Random Seeds

Unless otherwise stated, real-world performance refers exclusively to the multi-seed held-out test evaluation rather than isolated single-run validation outcomes. To assess statistical stability, the model was retrained across multiple random seeds using identical preprocessing, partitioning, and hyperparameter settings, thereby isolating variability attributable to stochastic initialization and data shuffling. Consequently, the observed dispersion reflects intrinsic training randomness rather than experimental inconsistency. Robustness was evaluated on a held-out test set, with variability characterized through standard deviation and confidence intervals (Table 7). Importantly, the discrepancy between single-run validation and multi-seed testing reflects expected generalization effects under unseen data. Although the CLAHE + sharpening configuration achieved competitive validation performance, this advantage was not consistently maintained during multi seed testing, likely due to stochastic optimization variability. Therefore, final performance is reported using multi-seed test averages to support reliable and reproducible deployment-oriented evaluation, while acknowledging that modest preprocessing differences may remain within the intrinsic stochastic behavior of the training pipeline.

Table 7. Performance stability across five random seeds. Values represent mean, standard deviation (Std.), and 95% confidence interval (CI). Validation and test metrics correspond to distinct evaluation protocols and therefore should not be interpreted interchangeably.

Metric	Mean	Std.	95% CI (Low–High)
Val mAP@50:95	0.64296	0.02252	0.61500–0.67092
Val mAP@50	0.79145	0.02673	0.75825–0.82464
Val Precision	0.82644	0.03083	0.78816–0.86471
Val Recall	0.72029	0.03469	0.67721–0.76337
Test mAP@50:95	0.61661	0.02728	0.58274–0.65048
Test mAP@50	0.77628	0.02719	0.74252–0.81005
Test Precision	0.77967	0.03860	0.73174–0.82759
Test Recall	0.73306	0.02862	0.69752–0.76860

The confidence intervals observed across validation and test metrics indicate stable convergence under stochastic variation. Test mAP@50:95 remains bounded across runs, supporting consistent generalization behavior. The relatively small dispersion observed across runs indicates that performance variability remains controlled despite stochastic training effects. While recall shows slightly higher variability than precision, no instability or performance collapse is observed. Figure 8 illustrates the relationship between validation and test performance, revealing comparable trends with a moderate generalization gap consistent with expected behavior under unseen data.

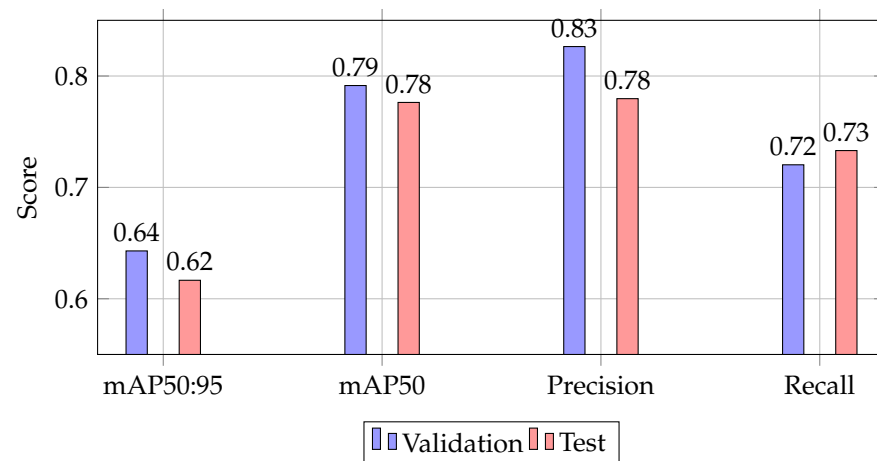


Figure 8. Comparison of validation and test performance across evaluation metrics. Results show comparable trends between datasets, with a moderate performance decrease consistent with expected generalization under unseen data.

4.5. Model and System-Level Performance

The inspection system performance is defined as the mean held-out test result across five random seeds, providing a statistically grounded estimate of generalization under stochastic variability. Accordingly, the averaged multi-seed test mAP@50:95 of approximately 0.617 is adopted as the primary deployment-oriented metric, whereas higher single-run validation results reflect only fixed initialization conditions and should not be interpreted as representative operational performance estimates. This design choice was motivated by the intention to prioritize reliable deployment behavior under realistic conditions. In addition, GPU processing remained below three seconds per image, with high job completion rates, stable large-scale visualization, and asynchronously processed georeferenced outputs enabled through YOLOv8n-based ingestion. Notably, CLAHE configurations did not maintain consistent performance under multi-seed evaluation. Per-class behavior should also be interpreted within the heterogeneous semantic roles of the detection categories, which include both defect states and operational reference structures. A detailed visualization of the interface is provided in Appendix A.

5. Discussion

The results of this study demonstrate that the primary novelty does not reside exclusively in detection performance, but rather in the integration of UAV-based defect detection, structured data persistence, and audited LLM-assisted querying within a unified offline system. By coupling statistically validated inspection models with constrained analytical interaction, the proposed framework redefines the inspection pipeline as an end-to-end, traceable, and operationally robust ecosystem under realistic field conditions. Consequently, the study shifts the emphasis from isolated predictive accuracy toward structured decision support and accountable analytical interaction. From this perspective, the principal contribution is architectural and methodological, motivated by the broader objective of translating computer vision capabilities into secure, interpretable, and deployment-oriented inspection workflows suitable for constrained operational environments.

5.1. Interpretation of Core Findings

As shown in Figure 8, the stability of mAP across seeds supports consistent behavior under stochastic variation and reinforces the reliability of the reported performance. The results demonstrate the feasibility of integrating UAV-based defect detection with an audited LLM-assisted analytics layer within a fully offline inspection ecosystem. Furthermore, the

stable multi-seed mean test mAP@50:95 supports reproducible generalization beyond favorable initialization conditions. Importantly, the study extends evaluation beyond predictive accuracy to examine operational behavior, including latency, batch reliability, and geospatial visualization aligned with practical deployment requirements. From this perspective, the contribution is primarily architectural and methodological, motivated by the objective of translating detection capability into operational value. Accordingly, the integration of computer vision inference, structured data persistence, and constrained querying within an offline-first design enables secure, traceable, and autonomous inspection workflows under realistic UAV acquisition conditions.

5.2. Comparison with Existing Approaches

Unlike prior approaches, which prioritize either detection accuracy or edge deployment, the proposed system explicitly integrates these dimensions within a unified and deployable framework. Prior work has explored UAV-based inspection and edge intelligence, primarily focusing on detection accuracy and real-time processing capabilities. For instance, UAV-driven infrastructure monitoring systems have demonstrated effective defect detection using deep learning models, yet they often rely on cloud-supported or semi-connected environments [3,10,14]. Similarly, edge intelligence frameworks emphasize low-latency processing, but typically lack integrated analytics layers and auditable data interaction mechanisms [19]. In contrast, this work extends beyond isolated detection performance by incorporating an offline-first architecture with structured persistence and constrained LLM-assisted querying, thereby addressing the gap between predictive capability and deployable, secure inspection workflows.

5.3. System and Deployment Implications

Accordingly, the reported performance metrics correspond exclusively to object detection evaluation rather than segmentation performance, while offline execution was prioritised to reduce operational risk, minimise latency, and avoid dependence on network connectivity during prolonged UAV inspection campaigns, particularly in remote or infrastructure-limited environments. Furthermore, offline execution reduces operational risk, latency, and dependence on network availability during extended UAV inspection campaigns, particularly in remote environments. By removing cloud reliance, the architecture improves resilience and supports continuous local processing and data access near the acquisition source. This design intentionally prioritizes field autonomy and robustness, reflecting a deployment-oriented research objective. At the same time, sustained local operation requires structured data lifecycle strategies, including supervised archiving and synchronization when connectivity is restored. To support stable performance, *Huey* enables asynchronous task orchestration, while SQLite, enhanced with indexing, Write-Ahead Logging (WAL), and periodic *VACUUM*, provides reliable embedded persistence. Nevertheless, prolonged large-scale deployments may require partitioning or hybrid synchronization mechanisms to maintain system stability.

5.4. Reliability and Trust

The proposed workflow emphasizes constrained and traceable analytical interaction through controlled query validation, structured schemas, and explicit result exposure. Audited text-to-SQL mechanisms support supervised data access using read-only constraints and schema-level validation, thereby reducing the risk of unintended or malicious operations. At the same time, the framework explicitly acknowledges inherent LLM limitations, including hallucination and non-deterministic reasoning, and mitigates them through transparent data exposure and user-level verification. Consequently, all generated outputs remain transparent to their underlying data sources, promoting accountable

decision-making within constrained operational settings. However, formal adversarial security guarantees, penetration testing, and certified robustness validation remain outside the scope of the present study. The notion of auditability adopted here therefore refers to operational traceability through observable and constrained query execution rather than formal cryptographic verification, reinforcing the system's role as a decision support framework in which human oversight remains essential.

5.5. Limitations and Future Work

Despite demonstrating functional feasibility, the present study remains constrained by factors affecting generalization and deployment readiness. The evaluation relies on a single-source dataset with limited environmental diversity, sensor variability, and class balance, while the relatively small number of unique base images restricts exposure to variations in illumination, acquisition altitude, weather conditions, infrastructure characteristics, and defect appearance distributions. Consequently, the reported findings should be interpreted within the evaluated acquisition context rather than as representative of broader operational environments. Although augmentation increased the effective training volume, the number of unique acquisition scenarios remained limited, and formal pairwise statistical comparisons between preprocessing configurations were not performed because observed differences fell within multi seed variability. Accordingly, future work will emphasize cross-dataset and cross-location validation using larger and more heterogeneous UAV inspection datasets, including the Chinese Power Line Insulator Dataset [31] and the Inspection Power Line Asset Dataset [32], to strengthen external validity, domain transferability, and field-ready generalization.

In addition, the absence of cross-architecture benchmarking and the reliance on a single model configuration may limit methodological breadth. From a systems perspective, the proposed framework should be interpreted as a validated prototype rather than a fully mature deployment solution, since scalability, adversarial robustness, and formal security verification were not comprehensively assessed. Although the implemented safeguards reduce unintended or destructive query execution through constrained analytical interaction, formal sandbox certification and adversarial validation remain outside the scope of the present study. Moreover, operational indicators, including execution latency, schema consistency violations, hallucinated attribute references, and adversarial query rejection rates, were not systematically quantified. The framework employs SQLite because of its portability and suitability for offline deployment; however, long-term database performance, including query latency, write throughput, and scalability, was not systematically evaluated, and future work should compare SQLite with alternative embedded analytical databases such as DuckDB. Furthermore, broader scalability studies and benchmarking against comparable integrated deployment frameworks remain necessary to strengthen deployment-oriented validation.

The present study evaluates the feasibility of supervised LLM assisted querying within an offline inspection workflow rather than benchmark level semantic parsing performance. The LLM assisted subsystem operates asynchronously outside the UAV flight control loop and therefore does not affect autonomous navigation or onboard stabilization processes. To support controlled analytical interaction, the framework incorporates offline embedded persistence, read only SQL enforcement, schema level validation, constrained query execution, and accountable logging. However, operational metrics, including query execution latency, hallucination frequency, schema consistency validation, adversarial query rejection, and user centered interaction quality, were not systematically quantified. Moreover, evaluating LLM generated outputs remains challenging because multiple semantically equivalent SQL formulations may produce identical results. Therefore, future work will

incorporate benchmark datasets, semantic validation protocols, LLM0 routing accuracy, SQL correctness, LLM3 acceptance and rejection rates, adversarial robustness assessment, multimodal integration, usability studies, and penetration testing to establish standardized, reliable, and secure evaluation of the LLM assisted analytical subsystem.

6. Conclusions

The reported deployment-oriented performance reflects statistically grounded multi-seed held-out testing rather than isolated validation outcomes obtained under fixed initialization conditions. Accordingly, the multi-seed held-out test performance should be interpreted as a more representative estimate of expected operational behavior than peak validation results derived from individual training configurations, as it better captures variability associated with initialization sensitivity and experimental stochasticity.

6.1. Summary of Core Findings

The proposed platform demonstrates the preliminary feasibility of combining offline computer vision with audited LLM-assisted analytics for infrastructure inspection. OffIn-spect LLM integrates visual signal processing and a constrained SQL interface within an offline-first architecture, evaluated on a realistic UAV dataset. The system achieved a competitive mAP@50:95 while maintaining per-image inference times compatible with field deployment. Beyond detection accuracy, the framework ensures traceability and incorporates practical security safeguards aligned with operational requirements. Importantly, its modular data model enables adaptation to alternative detection backbones and evolving domain schemas without structural redesign, reflecting a deliberate effort to design a secure and extensible inspection ecosystem grounded in real-world deployment constraints.

6.2. Architectural Contribution

This work demonstrates the feasibility of integrating UAV-based defect detection with audited conversational analytics within a fully offline inspection platform. Rather than prioritizing algorithmic novelty, the contribution is fundamentally architectural and methodological, focusing on secure system integration, controlled data interaction, and operational autonomy under field constraints. By coupling visual signal processing with constrained LLM-mediated querying, the framework establishes a deployable inspection infrastructure that emphasizes reliability, traceability, and resilience over isolated model performance improvements.

6.3. Practical Implications

Results suggest preliminary deployment feasibility under controlled operational conditions, providing inspectors with structured analytical outputs and interpretable decision support within realistic UAV inspection workflows. By integrating stable detection performance, low-latency processing, and auditable LLM-assisted querying, the system supports informed human oversight rather than autonomous replacement. This deployment-oriented validation suggests that the framework can function as a practical decision support tool in infrastructure inspection contexts while maintaining transparency and traceability.

6.4. Future Directions

Future research should investigate scalability, including variations in acquisition platforms, environmental conditions, infrastructure categories, and geographically distinct inspection scenarios, formal security benchmarking of LLM interactions, and the evaluation of lightweight detection backbones suitable for heterogeneous hardware environments. In addition, incorporating additional sensors and multimodal fusion strategies will enhance contextual awareness and defect discrimination in complex inspection scenarios.

Strengthening empirical validation through cross-architecture comparison and structured adversarial testing will clarify trade-offs between offline autonomy and cloud-based analytical capabilities. Ultimately, advancing formal verification mechanisms for LLM-generated queries remains essential to ensure robust security guarantees in long-term operational deployment. This work contributes toward bridging the gap between high-performing detection models and their reliable deployment in real-world inspection systems. Future work will incorporate curated natural language query benchmarks, semantic output validation strategies, execution safety metrics, and user-centered evaluation protocols to support systematic assessment of LLM-assisted analytical interaction. In particular, future studies should incorporate benchmark-driven semantic evaluation, human subject usability assessment, and quantitative query correctness analysis to rigorously validate the practical effectiveness of LLM-assisted analytical interaction within inspection environments.

Author Contributions: Conceptualization, R.V.; Writing—original draft, M.S. and R.V.; Writing—review and editing, M.S., R.V., P.O.-A. and J.V.; Project administration, J.V.; Funding acquisition, J.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
LLM	Large Language Model
CLAHE	Contrast Limited Adaptive Histogram Equalization
SQL	Structured Query Language
API	Application Programming Interface
mAP	Mean Average Precision
NMS	Non-Maximum Suppression
EXIF	Exchangeable Image File Format
WAL	Write-Ahead Logging
SGD	Stochastic Gradient Descent
GSD	Ground Sampling Distance
GPS	Global Positioning System
GUI	Graphical User Interface
GPU	Graphics Processing Unit
JSON	JavaScript Object Notation

Appendix A. User Interface and Application Views

This appendix presents representative interface views of the OffInspect-LLM platform to support transparency, reproducibility, and practical validation of the proposed system. The visualizations illustrate how UAV-acquired data are processed within an integrated offline pipeline that combines computer vision, geospatial analysis, and audited LLM-assisted querying. These examples provide insight into the interaction between system components, including inspection management, defect assessment, and controlled analytical querying, thereby reinforcing the feasibility of the architecture under real-world operational constraints. The appendix serves as complementary evidence of system integration and deployment readiness beyond the quantitative results reported in the main text.

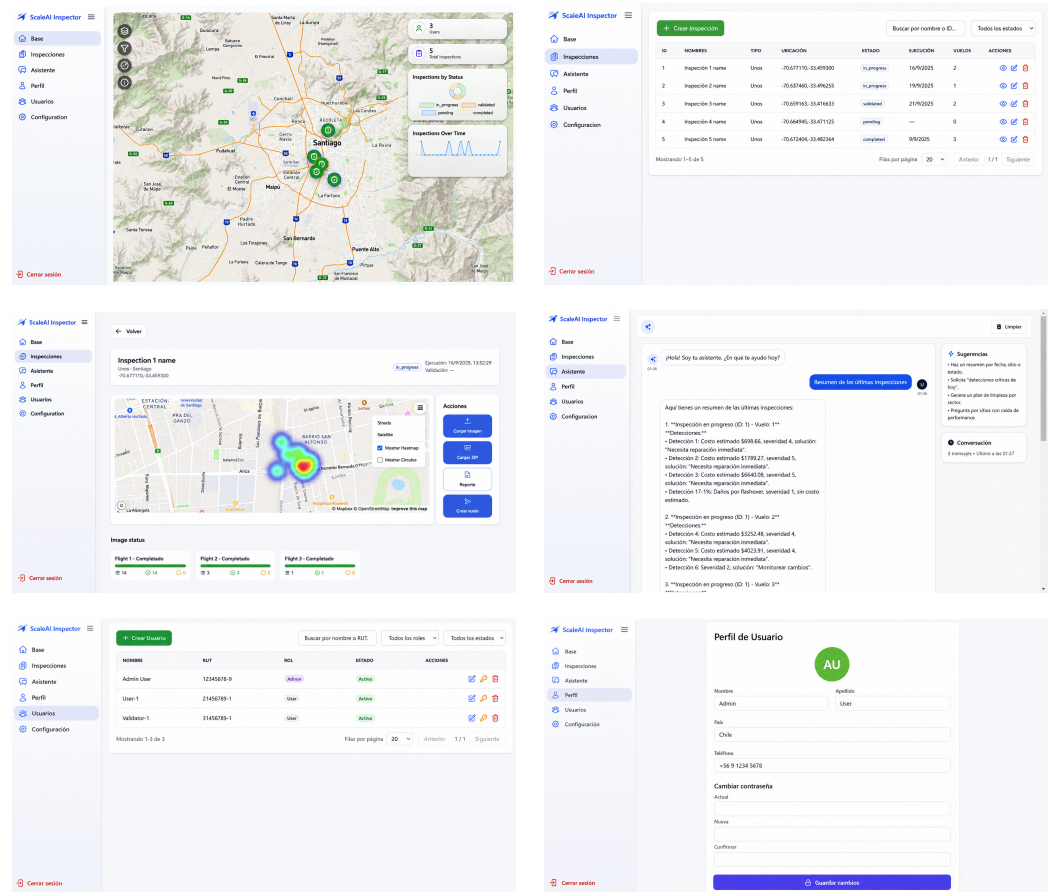


Figure A1. OffInspect-LLM refers to the proposed inspection platform rather than a standalone language model. The figure illustrates the integrated offline inspection workflow, including dashboard monitoring, inspection management, detailed defect visualization, audited conversational analytics, LLM-generated analytics, and user profile configuration.

References

1. Abu-Jassar, A.T.; Attar, H.; Amer, A.; Lyashenko, V.; Yevsieiev, V.; Solyman, A. Remote monitoring system of patient status in social IoT environments using amazon web services technologies and smart health care. *Int. J. Crowd Sci.* **2025**, *9*, 110–125. [\[CrossRef\]](#)
2. Tilon, S.; Nex, F.; Vosselman, G.; Sevilla de la Llave, I.; Kerle, N. Towards improved unmanned aerial vehicle edge intelligence: A road infrastructure monitoring case study. *Remote Sens.* **2022**, *14*, 4008. [\[CrossRef\]](#)
3. Wang, J.; Ueda, T.; Wang, P.; Li, Z.; Li, Y. Building damage inspection method using UAV-based data acquisition and deep learning-based crack detection. *J. Civ. Struct. Health Monit.* **2025**, *15*, 151–171.
4. Dhavalagimath, S.S.; Rajesh, T.; Singh, R.K. Design of an intelligent method for blur classification and adaptive restoration framework for robust aerial image deblurring operations. *Int. J. Inf. Technol.* **2026**, 1–10. [\[CrossRef\]](#)
5. Outay, F.; Mengash, H.A.; Adnan, M. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transp. Res. Part A Policy Pract.* **2020**, *141*, 116–129. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Wang, Y.; Xie, W.; Liu, H. Low-light image enhancement based on deep learning: A survey. *Opt. Eng.* **2022**, *61*, 040901. [\[CrossRef\]](#)
7. Lee, K.G.; Song, S.J.; Lee, S.; Yu, H.G.; Kim, D.I.; Lee, K.M. A deep learning-based framework for retinal fundus image enhancement. *PLoS ONE* **2023**, *18*, e0282416. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Meer, P. Robust techniques for computer vision. In *Emerging Topics in Computer Vision*; Prentice Hall: Hoboken, NJ, USA, 2004; pp. 107–190.
9. Lyu, Y.; Chen, J.; Song, Z. Image-based process monitoring using deep learning framework. *Chemom. Intell. Lab. Syst.* **2019**, *189*, 8–17. [\[CrossRef\]](#)
10. Lekidis, A.; Anastasiadis, A.G.; Vokas, G.A. Electricity infrastructure inspection using AI and edge platform-based UAVs. *Energy Rep.* **2022**, *8*, 1394–1411. [\[CrossRef\]](#)
11. Amin, S.U.; Hossain, M.S. Edge intelligence and Internet of Things in healthcare: A survey. *IEEE Access* **2020**, *9*, 45–59.

12. Shi, L.; Tang, Z.; Zhang, N.; Zhang, X.; Yang, Z. A survey on employing large language models for text-to-sql tasks. *ACM Comput. Surv.* **2025**, *58*, 54. [\[CrossRef\]](#)
13. Husom, E.J.; Goknil, A.; Astekin, M.; Shar, L.K.; KÅÿsen, A.; Sen, S.; Mithassel, B.A.; Soylu, A. Sustainable llm inference for edge ai: Evaluating quantized llms for energy efficiency, output accuracy, and inference latency. *ACM Trans. Internet Things* **2025**, *6*, 28. [\[CrossRef\]](#)
14. Sheraz, M.; Hussain, Z.; Abbas, S.; Masood, M.K.K.; Abdullah, M.; Zahid, Z.B. Autonomous UAV-based pipeline monitoring with YOLOv8 for real-time surface anomaly detection. *Measurement* **2026**, *267*, 120624. [\[CrossRef\]](#)
15. Shafique, A.; Cao, G.; Khan, Z.; Asad, M.; Aslam, M. Deep learning-based change detection in remote sensing images: A review. *Remote Sens.* **2022**, *14*, 871. [\[CrossRef\]](#)
16. Liu, J.; Xiang, J.; Jin, Y.; Liu, R.; Yan, J.; Wang, L. Boost precision agriculture with unmanned aerial vehicle remote sensing and edge intelligence: A survey. *Remote Sens.* **2021**, *13*, 4387. [\[CrossRef\]](#)
17. Khan, Z.; Nasir, A.; Mekid, S. Fault-tolerant control strategies for industrial robots: State of the art and future perspective on AI-based fault management. *Artif. Intell. Rev.* **2025**, *58*, 362. [\[CrossRef\]](#)
18. Todericiu, I.A. Virtual assistants: A review of the next frontier in AI interaction. *Acta Univ. Sapientiae Inform.* **2025**, *17*, 1. [\[CrossRef\]](#)
19. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [\[CrossRef\]](#)
20. Wang, Y.; Chen, K.; Tan, H.; Guo, K. Tabi: An efficient multi-level inference system for large language models. In Proceedings of the Eighteenth European Conference on Computer Systems, Rome, Italy, 8–12 May 2023; pp. 233–248.
21. Pichler, F.; Kern, B. Text-to-SQL for DataGovernance Technologies and Education. Ph.D. Thesis, OST Otschweizer Fachhochschule, Rapperswil-Jona, Switzerland, 2024.
22. Chafik, S.; Ezzini, S.; Berrada, I. Enhancing security in text-to-SQL systems: A novel dataset and agent-based framework. *Nat. Lang. Process.* **2025**, *31*, 1399–1422. [\[CrossRef\]](#)
23. Wu, T.; Miao, Z.; Huang, W.; Han, W.; Guo, Z.; Li, T. SGW-YOLOv8n: An improved YOLOv8n-based model for apple detection and segmentation in complex orchard environments. *Agriculture* **2024**, *14*, 1958. [\[CrossRef\]](#)
24. Pawlaszczyk, D.; Bodach, R.; Engler, P.; Kolouch, J.; Spranger, M.; Hummert, C.; Labudde, D. AI-based automated SQL query generation for SQLite databases in Mobile forensics. *Forensic Sci. Int. Digit. Investig.* **2026**, *57*, 302100. [\[CrossRef\]](#)
25. Taleb, I.; Guillaume, J.L.; Duthil, B. A survey on services placement algorithms in integrated cloud-fog/edge computing. *ACM Comput. Surv.* **2025**, *57*, 271. [\[CrossRef\]](#)
26. Kerle, N.; Nex, F.; Gerke, M.; Duarte, D.; Vetrivel, A. UAV-based structural damage mapping: A review. *ISPRS Int. J.-Geo-Inf.* **2019**, *9*, 14. [\[CrossRef\]](#)
27. Voron, F. *Building Data Science Applications with FastAPI*; Packt Publishing: Birmingham, UK, 2021.
28. Golpour, A.; Khanzadi, M.; Rahbar, M. Optimal UAV camera position for automated computer vision-based inspection of bolt looseness in steel structures based on 4D BIM. *J. Civ. Struct. Health Monit.* **2025**, *15*, 1557–1579. [\[CrossRef\]](#)
29. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [\[CrossRef\]](#)
30. Deng, N.; Chen, Y.; Zhang, Y. Recent advances in text-to-SQL: A survey of what we have and what we expect. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2166–2187.
31. Raimundo, A. *Insulator Data Set—Chinese Power Line Insulator Dataset (CPLID)*; IEEE Dataport: Piscataway, NJ, USA, 2020. [\[CrossRef\]](#)
32. Vieira e Silva, A.L.B.; Felix, H.d.C.; Simões, F.P.M.; Teichrieb, V.; dos Santos, M.; Santiago, H.; Sgotti, V.; Lott Neto, H. InsPLAD: A Dataset and Benchmark for Power Line Asset Inspection in UAV Images. *Int. J. Remote Sens.* **2023**, *44*, 7294–7320. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.