SJST
sjst.scst.edu.ly

Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)

مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

AI-Based Vision for Hydraulic-Structure Inspection: A DamCrack Feasibility Study

Rafea M. Almejrabi*, Ahmed Alwirshiffani, Mustafa Mohammed Alkharsh, Abdul Qadir Al-Tajouri, Fawzi Bou-Sha'ala, Nourah Mohammed Abdulmjid, Ramadan Ahmed M. Elghalid

College of Computer Technology, Benghazi, Libya
Corresponding author: rafeaalmejrab@gmail.com

Abstract

Reliable management of dams, spillways, canals, tunnels, and related hydraulic assets depends on timely visual evidence about deterioration. In many institutions, inspection is still based mainly on human observation, which may be slow, uneven between inspectors, and risky in locations that are high, confined, submerged, or difficult to access. This study develops a computer-vision framework for responsible AI-supported inspection of hydraulic structures. The framework links image capture, data screening, YOLO-style detection, U-Net-style segmentation, explainable outputs, and risk-based prioritization within a human-reviewed reporting process. The manuscript deliberately avoids unsupported claims of high accuracy and instead emphasizes reproducible data handling, annotation control, leakage-aware splitting, qualitative error review, and transparent reporting. To make the approach testable, DamCrack, a public drone and smartphone dataset for concrete-dam damage assessment, is selected as an initial validation source. A limited CPU-based trial was conducted on 500 DamCrack detection image-label pairs to verify end-to-end execution. On the held-out test split, the trial obtained precision = 0.386, recall = 0.402, mAP@0.5 = 0.340, and mAP@0.5:0.95 = 0.163. These values are reported as a feasibility baseline rather than as operational performance. They indicate that the workflow can be executed and audited, while larger data splits, GPU-based training, segmentation validation, and expert review remain necessary before field-deployment claims are made. The contribution is an inspection architecture and a practical validation route grounded in a real public dam-damage dataset.

Keywords: *Computer vision; Hydraulic structures; Concrete dams; DamCrack dataset; Deep learning; YOLOv8.*

المستخلص

تعتمد إدارة السدود والمفيضات والقنوات والأنفاق والمنشآت الهيدروليكية ذات الصلة على توفر أدلة بصرية موثوقة وفي الوقت المناسب حول مظاهر التدهور. تقدم هذه الدراسة إطاراً عملياً لفحص المنشآت الهيدروليكية بمساعدة الرؤية الحاسوبية والمخرجات، U-Net والتجزئة بأسلوب، YOLO من خلال ربط التقاط الصور، وضبط جودة البيانات، والكشف بأسلوب القابلة للتفسير، وترتيب الأولويات بناءً على المخاطر ضمن عملية مراجعة بشرية. ولا تدعي الدراسة جاهزية تشغيلية عالية بل تركز على قابلية إعادة الإنتاج، وضبط الترميز، وتقسيم البيانات، ومراجعة الأخطاء، والتقرير الشفاف. وللتحقق الأولي العامة الخاصة بتقييم أضرار السدود الخرسانية. أُجريت تجربة محدودة على DamCrack 500 استُخدمت مجموعة بيانات وحققت على مجموعة الاختبار دقة 0.386، واسترجاعاً 0.402، YOLOv8n زوج من الصور والوسوم باستخدام نموذج وتُعرض هذه القيم كخط أساس لإثبات جدوى سير العمل، لا .mAP@0.5:0.95 = 0.163 وmAP@0.5 = 0.340 كأداء تشغيلي نهائي.

Submitted: 10/05/2026 *Accepted: 27/06/2026*

1. Introduction

Hydraulic assets form part of the service backbone of modern communities. Dams, spillways, irrigation channels, intake works, water-conveyance tunnels, and similar facilities contribute to flood mitigation, water supply, agricultural production, power generation, and industrial operations. Because many such assets are exposed to long service periods and aggressive environments, systematic inspection is needed to document cracks, spalling, erosion, leakage indications, and other visible damage before they evolve into higher-risk conditions.

In practice, many inspections are still carried out through direct human observation. Inspectors may need to approach surfaces physically, use rope access, dive underwater, or inspect from boats or platforms. These approaches remain valuable because they allow professional judgment, but they can be costly, laborious, hazardous, and inconsistent. For large structures, remote sites, submerged regions, or high-elevation surfaces, purely manual inspection can also leave areas insufficiently documented.

UAVs, ROVs, high-resolution cameras, and deep-learning methods now make it possible to redesign parts of the inspection workflow. Vision models can screen large numbers of images, mark suspected defects, estimate affected regions, identify uncertain cases for review, and support maintenance documentation. Nevertheless, safe use of AI in hydraulic infrastructure cannot be reduced to a single accuracy number. It also requires controlled data preparation, defensible validation, baseline comparison, limitation reporting, and expert oversight when inspection outputs may influence safety-related decisions.

The present work therefore describes a reproducible inspection workflow and attaches it to a public, concrete validation route. Instead of claiming operational readiness without evidence, the study defines the data, annotation, evaluation, explainability, and governance conditions required before AI outputs are used in hydraulic-structure inspection. DamCrack is selected as the initial pilot dataset because it focuses on concrete dams and includes annotation resources for both detection and segmentation tasks [16].

The main contributions are as follows:

- A modular computer-vision framework for hydraulic-structure inspection that combines data acquisition, preprocessing, detection, segmentation, and decision-support layers.
- A dataset-grounded validation protocol that avoids unsupported claims by specifying dataset documentation, annotation quality checks, asset-level splitting, evaluation metrics, ablation analysis, and failure-case review.
- A risk prioritization model that links visual AI outputs to maintenance decision support through severity, area, location criticality, recurrence, and uncertainty indicators.
- A governance-oriented discussion showing how AI-assisted inspection should complement expert engineering judgment rather than replace formal structural assessment.

2. Background and Related Work

2.1 Visual inspection of hydraulic infrastructure

Hydraulic structures create difficult visual-inspection conditions. On exposed concrete, shadows, stains, moisture, vegetation, occlusion, texture change, and scale variation can reduce image clarity. Underwater imagery adds further difficulties, including turbidity, light scattering, color loss, and moving viewpoints. UAV and robotic inspection systems have therefore been investigated as ways to improve coverage and reduce the risks associated with close manual access. Previous work on autonomous underwater vehicles also shows that planned visual surveys of dam walls are technically feasible in submerged environments [1].

2.2 Deep learning for defect detection and segmentation

Deep learning is now widely used for image-based defect analysis. CNN models have been effective for concrete-crack recognition because they learn discriminative image patterns directly from data rather than relying entirely on manually designed features [2]. YOLO-family detectors are useful when the inspection objective is to locate many candidate defects efficiently with bounding boxes [3]. When more detailed spatial measurement is required, U-Net-type segmentation remains suitable because its encoder-decoder design and skip connections help preserve fine boundary information [4].

2.3 Data scarcity, augmentation, and imbalance

A persistent barrier is the limited availability of carefully labeled infrastructure-defect images. Damage classes may be infrequent, visually varied, and strongly imbalanced, which can weaken generalization, especially for rare conditions such as leakage, severe erosion, or underwater deterioration. Techniques such as focal loss can reduce the dominance of easy background or negative samples in dense detection settings [5]. Synthetic or domain-transfer augmentation, including CycleGAN-based approaches, may help only when generated images are reviewed for physical plausibility and evaluated through controlled experiments [6].

2.4 Explainability and decision support

For infrastructure management, AI outputs must be interpretable enough to support review rather than obscure it. Methods such as Grad-CAM can show which image regions contributed to a prediction and can help reveal attention to shadows, stains, or irrelevant background patterns [7]. In safety-sensitive inspection, predictions should be treated as screening evidence, not as final engineering judgments. This is why the proposed framework includes human verification, traceable reporting, and explicit uncertainty handling.

2.5 Research gap

Many published studies emphasize model performance for crack classification or defect localization, while fewer describe the complete path from image collection to institutional maintenance decisions. Important elements such as annotation governance, data splitting, segmentation, risk scoring, explainability, and reporting are often discussed separately from algorithmic evaluation. Reviews of crack detection, UAV/ROV inspection, and visual monitoring reflect this separation between model results and deployment procedures [10], [11], [14]. The framework proposed here addresses this gap by focusing on the full pathway from visual evidence to decision-support reporting.

3. Proposed Framework

The proposed workflow is structured as six connected layers: image acquisition, data-quality control and preprocessing, learning, explainable evidence generation, risk prioritization, and institutional reporting. The complete structure is summarized in Figure 1.

sjst.scst.edu.ly
Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)
مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

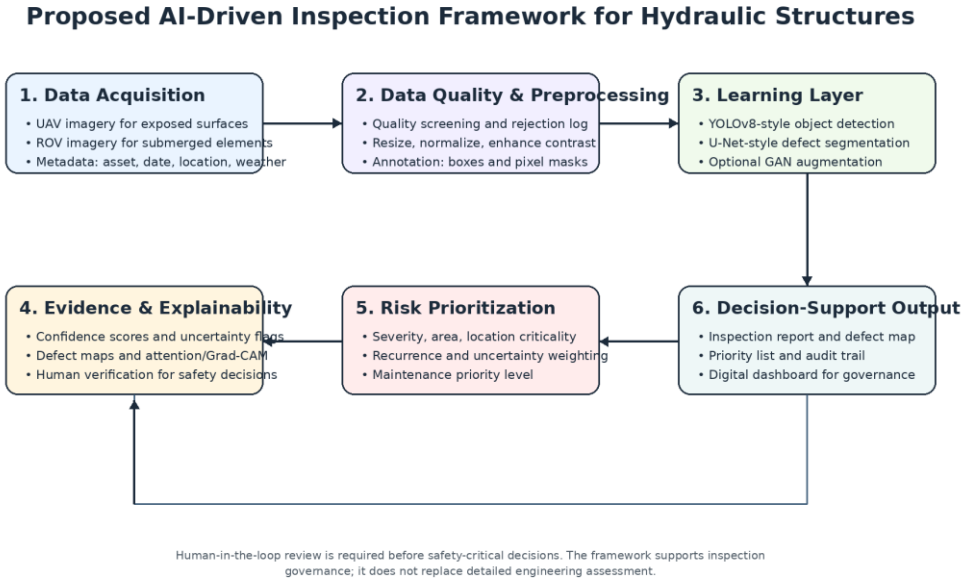

Figure 1. Proposed AI-driven inspection framework for hydraulic structures.

3.1 Defect categories and inspection evidence

The framework is designed to handle several visible defect types rather than a single crack-only task. Table 1 summarizes common categories and the evidence that should be recorded for each one.

Table 1. Defect categories and recommended evidence structure.

Defect category	Typical visual evidence	Recommended annotation	Operational concern
Cracking	Surface cracks or branching discontinuities on concrete	Box annotation plus mask when extent is needed	Possible durability loss and water-entry pathway
Spalling	Detached cover, exposed aggregate, or exposed reinforcement	Box or polygon/mask around the affected surface	Reduction of cover and protection of the concrete surface
Erosion/abrasion	Loss of material, roughening, or local scour marks	Region mask with estimated area or length	Hydraulic wear and weakening of the surface layer
Leakage/seepage	Wet marks, flow traces, discoloration, or deposits	Box annotation supported by context notes	Possible seepage route, joint issue, or internal flow indicator
Surface staining/biological growth	Color change, deposits, vegetation, algae	Classification flag or segmentation mask	May indicate moisture or maintenance need

3.2 Data acquisition layer

Images from UAVs are appropriate for visible surfaces such as dam faces, spillway slabs, canal linings, piers, and intake areas. Each flight should be documented with altitude, view angle, overlap, geolocation, asset code, weather, and illumination information. ROV images are more suitable for submerged components and should be accompanied by turbidity, depth, distance-to-surface, and route information.

For reproducible experiments, every image should carry metadata and a quality status. Images affected by strong blur, severe glare, heavy occlusion, or insufficient detail should be marked for exclusion or manual checking instead of being included in training without documentation.

3.3 Annotation and data governance

Labeling should follow a written annotation guide. Bounding boxes are adequate for object detection, while segmentation masks are required when the goal includes measuring defect area or boundaries. At least a reviewed subset should be labeled independently by two trained reviewers or experts so that disagreement and ambiguous cases can be identified.

When possible, data should be divided by asset or inspection session, not simply by individual image. This reduces the risk that near-duplicate frames from the same surface appear in both training and test sets and artificially improve the reported results.

3.4 Preprocessing and quality control

Preprocessing may involve resizing, intensity normalization, contrast improvement, and, for underwater images, optional color correction. The same preprocessing rules must be applied to the training, validation, and test partitions. Augmentation should imitate realistic capture variability, such as moderate rotation, scale differences, brightness changes, blur, and contrast variation.

A quality-control log should preserve rejected images and reasons for exclusion. This improves transparency and allows future auditing of the dataset creation process.

3.5 Object detection layer

A YOLOv8-like detector is suitable for screening large image collections because it can localize candidate defects and attach class labels and confidence values to each detection. For inspection use, confidence thresholds should be calibrated on validation data and reviewed with domain specialists rather than selected only to maximize a metric.

Imbalanced classes should be managed through sampling strategies, targeted augmentation, loss weighting, or focal-loss-type methods. Reports must include class-wise precision and recall because a high average score can hide weak detection of less frequent but important damage classes.

3.6 Semantic segmentation layer

A U-Net-like segmentation component is useful when the task requires boundary delineation or quantitative area estimation. This is relevant for measuring crack length, spalling extent, or eroded zones. Segmentation outputs can also feed the risk-prioritization layer by estimating the size and spread of a defect.

Segmentation quality should be reported using IoU and Dice coefficient at the defect-class level. Thin cracks require special attention because small annotation differences can produce large metric variation.

3.7 Optional generative augmentation layer

Generative augmentation can be considered, but only under explicit governance rules. CycleGAN-style translation may help represent different lighting conditions, surface textures, wetness levels, or underwater color shifts [6]. Such synthetic data should not be used as proof of field performance unless experts review the realism of generated samples and ablation results isolate their actual effect.

A separate experiment should compare training with and without synthetic data. The effect of synthetic augmentation must be reported through ablation analysis, not assumed.

SJST
sjst.scst.edu.ly
Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)
مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

3.8 Risk prioritization and reporting

The reporting layer translates visual findings into maintenance-oriented priorities. One possible normalized score combines severity, affected area, location criticality, recurrence, and uncertainty so that model outputs can be reviewed alongside engineering context.

$$Risk\ Score = 0.30(Severity) + 0.25(Area) + 0.20(Location\ Criticality) + 0.15(Recurrence) + 0.10(Uncertainty)$$

Each input is scaled from 0 to 100. The weighting scheme should be adjustable by an engineering committee according to asset type and institutional policy. The resulting score is a prioritization aid only; it is not a substitute for structural safety assessment.

Table 2. Operational scoring guide for risk-prioritization variables.

Variable	Purpose	Initial scoring rule	If data are unavailable
Severity	Represents visible seriousness of the defect	0-30 minor; 31-60 moderate; 61-100 severe or safety-relevant	Assign by expert review using an agreed inspection guide
Area / extent	Captures size or spread of the defect	Scale crack length, defect area, or affected proportion to 0-100	Use qualitative scale: small=25, medium=50, large=75, extensive=100
Location criticality	Accounts for where the defect appears on the asset	Assign higher values to spillways, intakes, joints, dam faces, and known stress zones	Use a committee-approved asset map; otherwise set neutral value of 50
Recurrence	Measures whether the issue appears repeatedly over time	0 for first observation; higher scores when repeated inspections show persistence or growth	Set to 0 or mark as unknown for the first inspection
Uncertainty	Penalizes low-confidence AI outputs or poor image quality	Higher values for low model confidence, blur, glare, turbidity, occlusion, or disagreement among reviewers	Require manual review and flag the case as uncertain

Table 3. Example risk prioritization matrix for AI-assisted inspection outputs.

Risk level	Score range	Recommended action	Decision authority
Low	0–39	Routine monitoring and periodic re-inspection	Maintenance team
Medium	40–59	Schedule engineering review and compare with previous inspection	Asset engineer
High	60–79	Prioritize field verification and maintenance planning	Senior engineer / inspection committee
Critical	80–100	Immediate engineering assessment and safety decision protocol	Responsible authority

sjst.scst.edu.ly
Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)
مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

4. Validation Protocol

Experimental reporting should be based on a documented validation protocol rather than unsupported claims. Table 4 lists the minimum checks that should be completed before reporting model performance as scientific evidence.

Table 4. Validation protocol required before reporting experimental performance claims.

Validation element	Required practice	Recommended metrics	Reason
Dataset documentation	Describe source, asset, image count, resolution, capture conditions, and license	Dataset card	Allows other researchers to understand and repeat the experiment
Annotation quality	Use a written labeling guide and review a sample independently	Agreement rate, Cohen/Fleiss kappa where applicable	Controls label noise
Detection evaluation	Test on independent assets or clearly state when this is not possible	Precision, recall, F1, mAP@0.5 and mAP@0.5:0.95	Measures localization and classification
Segmentation evaluation	Use manually verified masks	IoU, Dice, boundary error	Measures defect extent estimation
Ablation analysis	Remove or replace components one at a time	Metric deltas with confidence intervals	Shows contribution of each component
Failure-case review	Inspect false positives and false negatives	Error categories and examples	Improves field reliability
Operational test	Evaluate processing time and workflow usability	FPS, report generation time, expert review time	Connects model output to real inspection process

4.1 Recommended experimental design

A sound experiment should separate training, validation, and testing data in a way that limits leakage. When images originate from overlapping UAV frames, the same flight path or surface zone should not be split across train and test partitions. If asset-level metadata are unavailable, this limitation must be declared explicitly.

The experimental design should include baseline comparisons, such as simple image-processing rules, a lightweight CNN classifier, YOLO-only detection, U-Net-only segmentation, and, when available, the combined detection-segmentation workflow. Ablation results should show which component is responsible for any improvement.

The results should include qualitative cases, not only aggregate metrics. Correct detections, missed defects, false alarms, and low-confidence predictions are especially important for engineering readers because they reveal whether the model fails in safety-relevant ways.

4.2 Illustrative feasibility pilot

Before presenting broad performance conclusions, the workflow should be tested through a limited, auditable pilot. Public concrete-damage datasets can support this stage if the source, labeling scheme, split logic, and exclusion rules are stated clearly.

A minimum pilot should provide a dataset description, annotation guidance for boxes and masks, a baseline model, quantitative metrics, example predictions, an error log, and a short statement explaining what the pilot can and cannot prove.

Table 5. Minimum evidence package for an illustrative pilot study.

Pilot element	Minimum requirement	Publication value
Image subset	50-200 images from a clearly documented source with leakage control where possible	Shows that the pipeline can run end-to-end
Annotation subset	Bounding boxes for detection and masks for at least part of the data	Demonstrates compatibility with YOLO-style and U-Net-style workflows
Baseline model	Simple detector/segmenter or pretrained model fine-tuning	Prevents a purely theoretical presentation
Visual results	At least 6-10 panels showing correct detections, missed cases, false alarms, and uncertainty	Supports qualitative engineering interpretation
Expert review	Short notes from at least one civil/infrastructure reviewer	Connects AI output to field relevance

4.3 Public dataset selection: DamCrack

This manuscript strengthens the framework by connecting it to a specific public dataset. DamCrack contains drone and smartphone images from an operational concrete dam in Spokane, Washington, USA, and documents real damage patterns, mainly cracks and spalling. Its available annotations support classification, object detection, and segmentation [16]. This makes it more relevant to hydraulic-structure inspection than generic crack datasets collected from pavements or buildings.

DamCrack is suitable for this study for three main reasons. It is associated with a dam environment, it provides labels compatible with YOLO-type detection and U-Net-type segmentation, and it includes both aerial and close-range imagery. These characteristics align well with the proposed acquisition and analysis workflow [16].

Any use of DamCrack should follow the Zenodo CC BY 4.0 license. Researchers should cite the Data in Brief article and the dataset DOI, and should clearly state that the data are public secondary research data rather than images collected by the present authors.

The dataset mainly covers cracks and spalling. Other defect types considered by the framework, such as erosion, leakage, abrasion, and reinforcement exposure, require additional datasets or future field acquisition before they can be validated empirically.

Table 6. DamCrack dataset suitability for preliminary validation of the proposed framework.

Feature	DamCrack characteristic	Use in this paper
Infrastructure type	Operational concrete dam and spillway imagery	Consistent with the hydraulic-infrastructure scope
Damage types	Cracks and spalling	Covers the two visible-defect classes used in the present pilot
Image sources	Drone imagery and smartphone imagery	Supports UAV-based and close-range inspection scenarios.
Dataset scale	11,288 images, including 1,288 high-resolution aerial images, 6,000 classification patches, and 4,000 annotated detection/segmentation patches	Sufficient for a small preliminary pilot without collecting new field data.
Annotation formats	YOLO TXT labels, Pascal VOC XML, and PNG segmentation masks	Directly supports YOLO-style detection and U-Net/segmentation experiments.
Research use	Published with public repository access and Creative Commons Attribution 4.0 International (CC BY 4.0) data license on Zenodo	Allows transparent citation and reproducible pilot design, provided the original dataset and article are properly cited.

4.4 Proposed preliminary validation using DamCrack

The next experimental step is a bounded feasibility study using the DamCrack detection and segmentation components, not a claim of final inspection accuracy. The image or patch should be treated as the unit of analysis, with explicit records of the split procedure, model settings, and outputs.

The YOLO label files can support a detection baseline for cracks and spalling. The mask files can support a segmentation baseline using U-Net-like or YOLO-seg-like models. The two outputs can later be combined to estimate location, confidence, and affected area for risk scoring.

Table 7. Dataset-grounded pilot design for DamCrack-based validation.

Step	Input	Model / method	Output evidence to report	Status
Dataset card	DamCrack repository and article metadata	Document source, license, tasks, classes, annotation formats, exclusion rules	Dataset card included in appendix or supplementary file	Completed
Detection pilot	640 x 640 annotated patches with YOLO labels	YOLOv8 for crack and spalling detection	Class-wise Precision, Recall, F1, mAP@0.5, mAP@0.5:0.95, and sample predictions	Completed
Segmentation pilot	PNG masks for crack and spalling regions	U-Net-style or YOLO-seg segmentation	IoU, Dice, sample masks, boundary-error examples	Proposed
Ablation	Baseline plus optional	Compare without/with	Small ablation table, not exaggerated claims	Proposed

	prepro-cessing/augmen-tation	augmentation and quality filtering		
Risk mapping	Predicted defect class, area, confi-dence, and loca-tion metadata where available	Apply the risk-prioritization equation as a re-porting layer	Example risk re-port for selected images	Proposed
Failure review	False positives, missed fine cracks, ambigu-ous spalling, shadows/stains	Expert review checklist	Qualitative error analysis section	Proposed

4.5 Preliminary feasibility experiment using DamCrack

The pilot notebook was executed to check whether the workflow could run from data retrieval to model evaluation. The aim was feasibility, not performance optimization. The DamCrack Damage Detection subset was downloaded, extracted, converted to YOLO format, and used to train a baseline detector. The extracted subset contained 4,000 images and 4,000 matching label files. Using a fixed random seed, 500 image-label pairs were selected and divided into 350 training, 75 validation, and 75 test images. The labels consisted of two classes: crack and spall-ing.

A pretrained YOLOv8n model was fine-tuned for 10 epochs at 640-pixel input size with batch size 8. The short training schedule was chosen to keep the CPU-only run feasible, so full con-vergence was not expected. Ultralytics automatically selected AdamW with an initial learning rate of 0.001667 and momentum of 0.9. Because the run used CPU rather than GPU, the results are best interpreted as a lower-bound baseline for workflow feasibility, not as a comparison with optimized deep-learning studies. The trained model was then evaluated on the held-out test split.

Table 8. Initial DamCrack YOLOv8n feasibility-pilot results on the held-out test split.

Class	**Images / images contain-ing class**	**Total de-fect in-stances**	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
Overall	75	880	0.386	0.402	0.340	0.163
Crack	70	778	0.416	0.442	0.389	0.184
Spalling	49	102	0.356	0.363	0.291	0.141

Note. Some test images include both crack and spalling annotations; therefore, the per-class image counts can exceed the total number of test images. Precision and recall are taken from the Ultralytics/YOLO test output, whereas mAP summarizes performance across confidence thresholds.

The results demonstrate that the pipeline can be executed, but they also show that the current baseline should not be used for operational claims. The relatively low scores are consistent with the restricted setup: a small subset, 10 training epochs, the nano model variant, CPU-only exe-cution, and a simple random split rather than a full asset- or session-level split. The F1-confi-dence curve reached an all-class F1 of about 0.39 near a confidence threshold of 0.161. Future

experiments should therefore use larger splits, GPU training, tuned hyperparameters, segmentation evaluation, and expert review of false positives, missed defects, and boundary errors.

4.6 Qualitative diagnostics and reporting rule for preliminary results

The pilot should be described as a feasibility demonstration. A scientifically appropriate statement is: 'A preliminary experiment using the public DamCrack dataset shows that the proposed workflow can be executed and evaluated.' It would be inappropriate to state that the system is ready for operational dam-safety inspection, because such a claim would require field validation, cross-site testing, independent review, and formal engineering approval.

Only the object-detection part of the protocol was executed in the present pilot. Segmentation, ablation testing, risk-score mapping, and expert validation remain planned steps, as indicated in Table 7. Figures 2-4 therefore present detection examples and diagnostic plots only; they support transparent error analysis but do not imply deployment readiness.

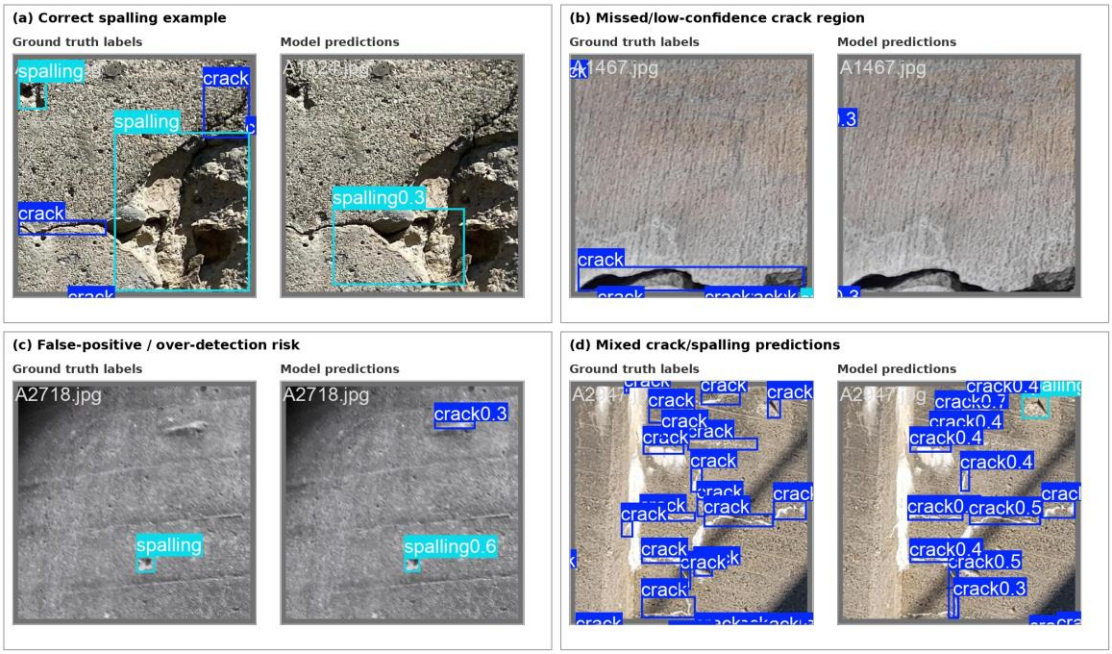

Figure 2. Qualitative examples from the DamCrack detection pilot. Each panel compares ground-truth labels with model predictions for a representative validation image tile, illustrating correct detection, missed or low-confidence regions, false-positive or over-detection risk, and mixed crack/spalling behavior.

sjst.scst.edu.ly

Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)

مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

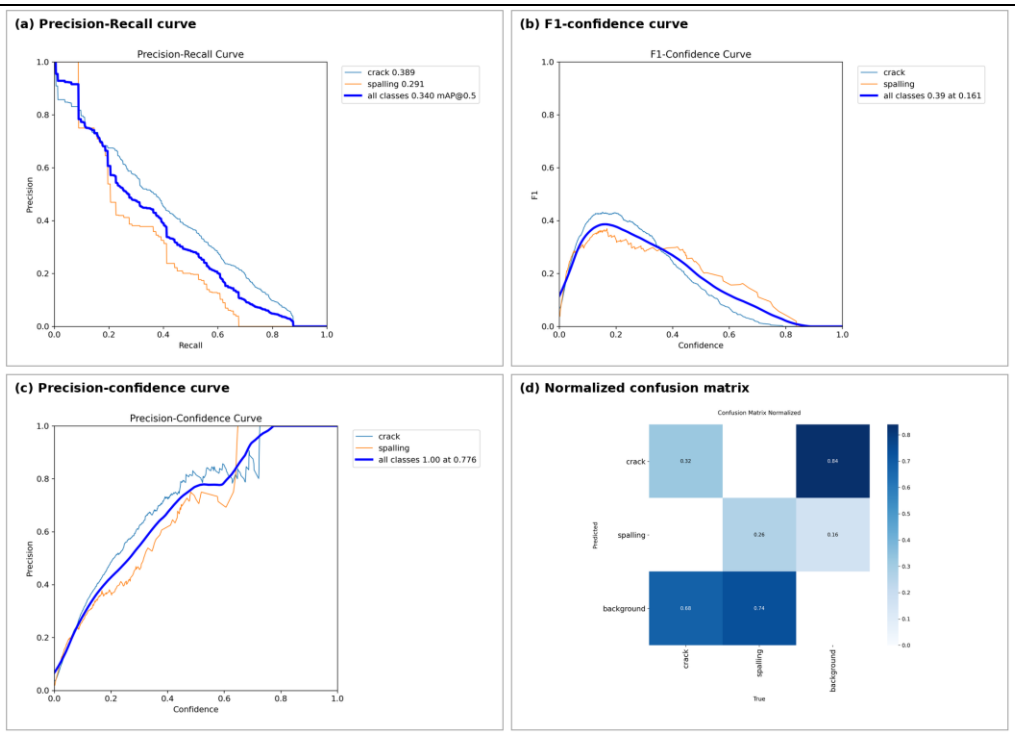

Figure 3. Diagnostic plots from the YOLOv8n DamCrack feasibility run: precision-recall behavior, F1-confidence behavior, precision-confidence behavior, and normalized confusion matrix. These diagnostics support transparent interpretation of the lower-bound baseline.

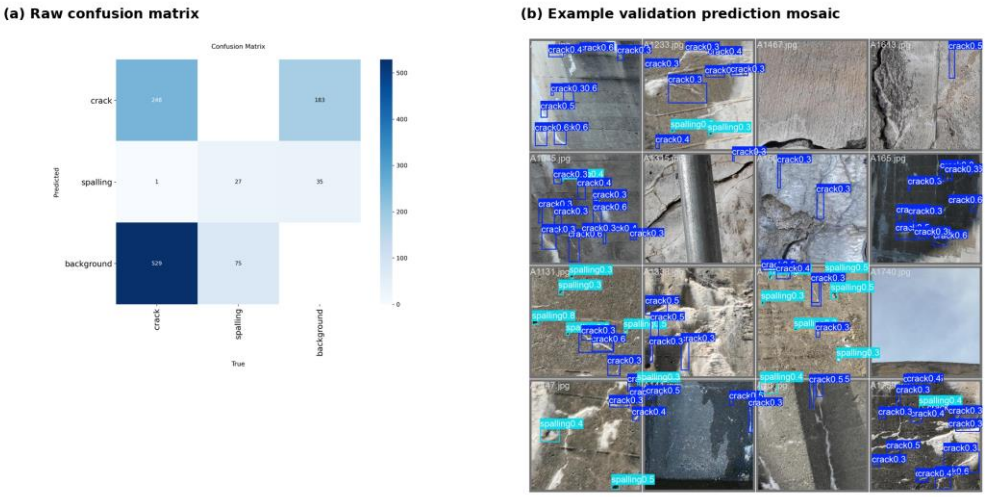

Figure 4. Raw confusion matrix and example validation prediction mosaic from the DamCrack detection pilot. The visual output shows both useful detections and over-detection/failure patterns that require expert review before operational use.

SJST
sjst.scst.edu.ly
Surman Journal for Science and Technology
ISSN: Online (2790-5721) - Print (2790-5713)
مجلة صرمان للعلوم والتقنية
Vol 8, No.2. Jun – Dec. 2026
Pages: 113 ~ 128

5. Implementation Pathway for Institutions

The framework can be implemented in three progressive phases rather than as a single high-risk deployment.

Table 9. Progressive implementation pathway for adopting AI-assisted inspection.

Phase	Objective	Main activities	Output
Phase 1: Documentation	Create reliable inspection dataset	Collect or select images, define labels, and verify annotation rules	Dataset card and annotation guide
Phase 2: Model validation	Assess AI feasibility	Train baseline/deep models, compute metrics, and inspect errors	Validation report and model card
Phase 3: Decision support	Integrate with maintenance workflow	Risk scoring, dashboard, expert verification, report generation	Inspection dashboard and audit trail

5.1 Indicative timeline and resource estimate

Institutions that wish to adopt the framework should begin with a small and auditable pilot. The schedule below assumes a limited dataset and two reviewers. It should be adjusted according to asset size, image volume, inspection complexity, and the availability of engineering reviewers.

For a pilot of roughly 1,000 images, the workload is expected to be about 30-40 person-days, including dataset organization, annotation review, baseline training, error analysis, and expert validation. Additional time is needed when pixel masks, multiple defect classes, or underwater imagery are included.

Table 10. Indicative implementation timeline for a small pilot deployment.

Stage	Typical duration	Main resources	Decision gate
Dataset preparation	two to four weeks for about 1000 to 2000 images	Public or newly acquired images; annotation software; two reviewers	Dataset card and annotation guide approved
Pilot training and testing	one to two weeks	GPU/cloud notebook; baseline detector; documented train-validation-test split	Metrics and failure cases reviewed
Expert validation	one week	Civil or infrastructure reviewer; inspection checklist	Model outputs accepted for screening only
Reporting prototype	one week	Report template; defect map; risk matrix	Maintenance-priority report generated

5.2 Human-in-the-loop governance

Because hydraulic infrastructure is safety-critical, AI outputs should be checked before they affect maintenance or safety decisions. The system should highlight low-confidence predictions, inconsistent detection/segmentation results, and defects in high-criticality zones. Final records should connect image evidence, model output, expert verification, and recommended action.

5.3 Reporting requirements

An inspection report should record the asset, date, capture method, image coverage, defect maps, class labels, confidence values, risk levels, reviewer notes, and recommended follow-up. It should also state clearly that AI screening supports but does not replace formal engineering assessment.

6. Discussion

The value of the framework lies in linking deep-learning outputs to inspection governance. Algorithm-focused studies often stop at performance metrics, while practical inspection requires decisions about review, prioritization, reporting, and accountability. Here, detection and segmentation are treated as evidence-producing steps within a wider management workflow.

The framework is useful for post-event screening, routine monitoring of broad concrete surfaces, comparison of repeated image campaigns, and ranking locations that need closer engineering assessment. It can also reduce unnecessary exposure of inspectors to hazardous access zones by allowing image review before detailed field visits are planned.

However, the workflow must not be read as a replacement for structural diagnosis. A visible crack or spalled region may indicate a deeper process that requires engineering analysis, non-destructive testing, hydrological information, or material investigation. Computer vision is therefore best used for early flagging, documentation, and prioritization.

7. Limitations

The study remains implementation-oriented: it identifies a public dataset and reports an initial detection pilot, but the results are not sufficient for operational performance claims. Broader claims require larger training runs, independent test sets, prediction examples, and external review.

Transferability to other structures, materials, countries, cameras, illumination conditions, and underwater environments still needs empirical testing.

Defect appearance does not always imply structural severity; engineering interpretation remains necessary.

Synthetic images generated through GAN-based augmentation may introduce unrealistic artifacts unless reviewed by domain experts.

Regulatory adoption requires documentation, human oversight, failure-case analysis, and clear accountability procedures.

8. Conclusion

This paper developed a reproducible framework for AI-assisted visual inspection of hydraulic structures. The proposed workflow combines UAV/ROV image acquisition, image-quality control, defect detection, segmentation, explainability, and risk-based prioritization under human supervision. The framework was linked to the public DamCrack dataset and tested through a limited CPU-based YOLOv8n feasibility run using 500 detection image-label pairs. Quantitative metrics, confusion matrices, and prediction mosaics were reported to make the pilot auditable. The test outcome (overall mAP@0.5 = 0.340; mAP@0.5:0.95 = 0.163) should be read as evidence of workflow execution rather than as evidence of field readiness. The modest baseline performance highlights the need for larger splits, GPU optimization, segmentation testing, leakage-aware partitioning, and civil-engineering review before any dam-safety use is proposed. The main contribution is a transparent pathway from image capture to AI-assisted prioritization and evidence-based reporting.

Appendix A. Recommended DamCrack Pilot Notebook

The supplementary cloud notebook downloaded the DamCrack detection subset, constructed a small YOLO-format split, trained a YOLOv8n baseline, and exported metrics, prediction mosaics, confidence curves, precision-recall curves, and confusion matrices. In the logged run, 4,000 images and 4,000 label files were detected; 500 image-label pairs were selected and divided into 350 training, 75 validation, and 75 testing images. The configuration used 640-pixel input size, batch size 8, 10 epochs, seed 42, CPU execution, and the Ultralytics auto-selected AdamW optimizer (initial learning rate 0.001667, momentum 0.9). Future notebook versions should include full-subset training, GPU execution, segmentation masks, automated export of representative examples, and a structured failure-case sheet.

ACKNOWLEDGEMENTS

The authors acknowledge the public availability of the DamCrack dataset and the documentation provided by its creators, which supported the preliminary feasibility validation reported in this study.

DECLARATION OF CONFLICTING INTERESTS

The author(s) declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

FUNDING

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Declaration of Generative AI and AI-assisted technologies in the writing process

In preparing this manuscript, the authors used ChatGPT (OpenAI) for language polishing, organization of manuscript sections, formatting assistance, and readability support. All scientific claims, dataset descriptions, references, experimental outputs, and interpretations were checked and edited by the authors. The final article was prepared with substantial human review, and the authors accept full responsibility for its content.

References

[1] Ridao, P., Carreras, M., Ribas, D., & Garcia, R. (2010). Visual inspection of hydroelectric dams using an autonomous underwater vehicle. Journal of Field Robotics, 27(6), 759–778. https://doi.org/10.1002/rob.20351

[2] Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361–378. https://doi.org/10.1111/mice.12263

[3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788. https://doi.org/10.1109/CVPR.2016.91

[4] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

[5] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision, 2999–3007. https://doi.org/10.1109/ICCV.2017.324

[6] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision, 2223–2232. https://doi.org/10.1109/ICCV.2017.244
[7] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision, 618–626. https://doi.org/10.1109/ICCV.2017.74
[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778. https://doi.org/10.1109/CVPR.2016.90
[9] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965
[10] Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal, 57(2), 787–798. https://doi.org/10.1016/j.aej.2017.01.020
[11] Zhu, Y., & Tang, H. (2023). Automatic damage detection and diagnosis for hydraulic structures using drones and artificial intelligence techniques. Remote Sensing, 15(3), 615. https://doi.org/10.3390/rs15030615
[12] Yaseen, M. (2024). What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. arXiv preprint arXiv:2408.15857.
[13] Ultralytics. (2023). Ultralytics YOLOv8 model documentation. Retrieved from https://docs.ultralytics.com/models/yolov8/
[14] Capocci, R., Dooly, G., Omerdić, E., Coleman, J., Newe, T., & Toal, D. (2017). Inspection-class remotely operated vehicles—A review. Journal of Marine Science and Engineering, 5(1), 13. https://doi.org/10.3390/jmse5010013
[15] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27.
[16] Gharehbaghi, V., Bennett, C., Lequesne, R. D., Zhao, H., & Li, J. (2026). DamCrack: A drone and smartphone image dataset for 2D and 3D damage assessment in concrete dams. Data in Brief, 66, 112737. https://doi.org/10.1016/j.dib.2026.112737; dataset repository: https://doi.org/10.5281/zenodo.17274707