Cracks in the Foundation: A Civil Infrastructure Dataset to Challenge Vision Foundation Models

Nicola Farronato*
IBM Research
ETH Zürich
nicola.farronato@ibm.com

Niccolo Avogaro*
IBM Research
ETH Zürich
niccolo.avogaro1@ibm.com

Thomas Frick
IBM Research

Mattia Rigotti
IBM Research

Rizwan Ullah Khan
University of Twente

Michele Magno
ETH Zürich

Konrad Schindler
ETH Zürich

Cristiano Malossi
IBM Research

Florian Scheidegger
IBM Research

Abstract

Automated structural health monitoring is essential to prevent catastrophic infrastructure failures. Precise, pixel-level defect segmentation is needed to accurately assess structural integrity, but progress in defect segmentation for civil infrastructures has been held back by an extreme scarcity of data, which requires costly expert annotation. The need for data is accentuated by algorithmic hurdles intrinsic to the problem, including center-bias and the need to rely more on shape when inspecting nearly textureless building materials. To remove the bottleneck, we introduce **Cracks in the Foundation (CiF)**, the largest and most detailed civil infrastructure (instance) segmentation dataset to date, comprising ≈150,000 high-resolution tiles meticulously curated over five years in collaboration with civil engineering experts. With the help of this unprecedented data source, we expose a blind spot of current visual AI: despite the advent of promptable Foundation Models (FMs) and Vision Language Models (VLMs), and despite the impressive abilities of today's specialised segmentation models, it turns out that dense image understanding in the built environment is nowhere near solved. Our evaluations indicate that even the most recent zero-shot FMs face significant challenges when deployed on real-world infrastructure and even the performance of specialised models with domain-specific supervision plateaus at ≈25% mAP. CiF establishes inspection of civil infrastructure, an elementary and seemingly easy perceptual task, as an open challenge that reveals fundamental weaknesses of present-day models trained predominantly on internet images, literally and figuratively highlighting *cracks in the current foundation model paradigm*. Dataset available at: https://huggingface.co/datasets/ibm-research/cif-dataset

1 Introduction

Automated structural health monitoring stands to revolutionize the maintenance of global civil infrastructure [18, 36]. In this domain, undetected defects can lead to catastrophic failures that cause massive economic damages and seriously harm humans. To assess the integrity of a structure, civil engineers require precise quantification of damage type, area, shape, and distribution. Technically,

*Equal Contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

mapping (visible) structural damages amounts to a pixel-level segmentation problem. Given the high stakes and the complex physics of materials and their failure, high accuracy is required before a system can be deployed in the real world.

Training as well as evaluation of segmentation methods for civil infrastructure have been hobbled by data scarcity, and complicated by algorithmic peculiarities compared to general-purpose image segmentation [2]. Datasets at the scale needed for contemporary machine learning are lacking, as crowdsourcing is not practical when annotators must have the expertise to distinguish microscopic deterioration from harmless shadows. Yet, it is not just a data problem: we will show that even with abundant data, the domain exposes fundamental weaknesses in modern visual AI architectures: the monochromatic, flat appearance of concrete means that understanding must rely on shape rather than texture, a notorious weakness of deep learning models [20]. The issue is further exacerbated by thin, highly non-convex target structures incompatible with the models' known center-bias [52].

To plug this critical gap, we introduce Cracks in the Foundation (CiF), illustrated in Figure 1, a civil infrastructure dataset with instance annotations that is the largest of its kind in terms of both volume and resolution, consisting of ≈150,000 tiles and ≈250,000 fine-grained masks. Curated and meticulously annotated over five years of direct collaboration between AI industry partners and civil engineering experts, CiF captures the full scale range and visual complexity of infrastructure "in the wild." By open-sourcing this unprecedented collection, we provide a rigorous, analysis-ready benchmark that exposes the limits of modern vision models. We hope to challenge the community and to ultimately foster the development of visual AI capable of safeguarding our physical environment.

Cracks in the Foundation especially point at an issue that has been largely overlooked in the recent AI boom. The advent of Vision Language Models [1, 34, 6, 32, 37] (VLMs), especially those capable of visual grounding [41, 16, 50, 42, 30, 7, 14], and of promptable Foundation Models [27, 43, 12, 35] (FMs) has to some degree created the illusion that dense image understanding is solved [5]. Applying these models to the civil infrastructure domain reveals a serious, unexpectedly strong limitation: zero-shot generalization falls remarkably short. Cutting-edge FMs, trained predominantly on internet-scraped image collections like COCO [33] or LAION [47], seem to lack understanding of the features and scale variations needed in civil engineering applications. Experiments with our benchmark demonstrate that state-of-the-art promptable models fail to identify structural anomalies, and even domain-specific, supervised models plateau at ≈25% mAP. In other words, segmenting civil infrastructure degradations is, despite its straightforward nature, an open problem for present-day visual AI and challenges current solutions, literally revealing *cracks in the foundation* model paradigm.

2 Related Works

Large-scale benchmarks have catalyzed modern computer vision [45, 47], with dense prediction datasets like MSCOCO [33], ADE20K [53], and SA-1B [27] demonstrating that high-quality instance masks are essential for training robust object detection and segmentation architectures. However, civil infrastructure inspection remains a highly specialized outlier. In this domain, generic features consistently fail to capture critical, fine-grained anomalies—such as subtle crack patterns or complex concrete deterioration—under variable real-world conditions.

To address these domain-specific needs, the civil infrastructure landscape has seen a proliferation of datasets, as comprehensively surveyed by Bianchi and Hebdon [10], who cataloged 86 datasets up to 2022. These resources have evolved through levels of increasing annotation complexity, although the vast majority remain restricted to classification or semantic segmentation tasks. Early efforts primarily provided image-level or patch-level labels; for example, CDS [22] and SDNET2018 [17] focused on binary crack detection in concrete patches. More comprehensive multi-class datasets like MCDS [23] and multi-label benchmarks like CODEBRIM [38] expanded these to multiple reinforced concrete defect types. Pixel-level semantic segmentation has further matured this landscape, with single-class crack datasets like UAV75 [8] and CrackSeg9k [28] providing large-scale binary masks. Recent state-of-the-art resources such as S2DS [9] and dacl10k [19] represent the current pinnacle of this progression, with dacl10k offering 9,920 images across 19 classes for semantic bridge damage segmentation.

Despite this growth, instance segmentation remains significantly underdeveloped in the civil infrastructure landscape. While semantic segmentation identifies defect types, it does not distinguish

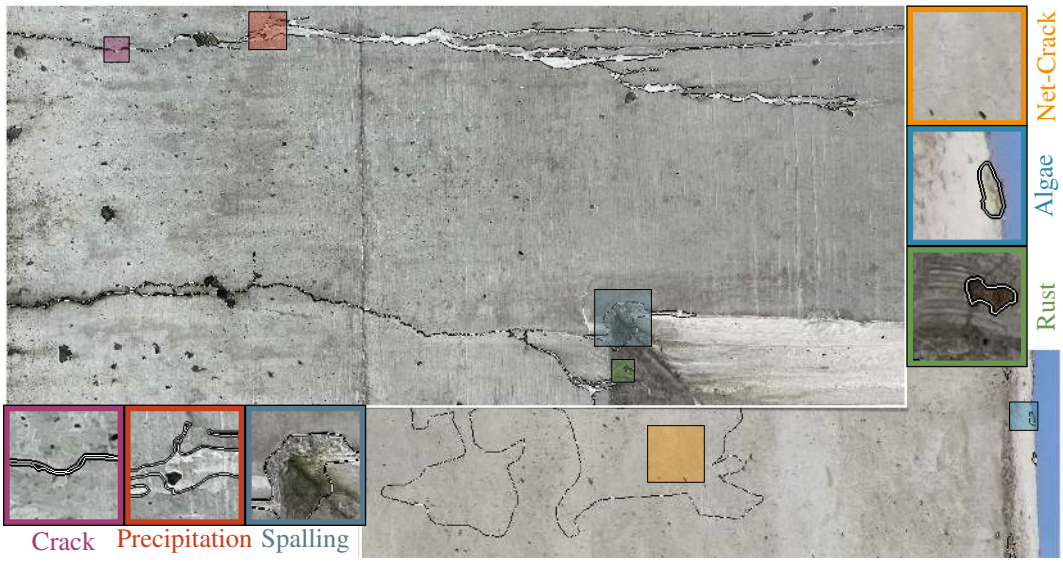


Figure 1: Representative high-definition images from Cracks in the Foundation dataset. Due to the high resolution of the imagery and the fine scale of the structural anomalies, magnified callouts are utilized to visualize specific defect classes: Crack (magenta), Precipitation (red), Spalling (blue), Net-Crack (orange), Algae (cyan), and Rust (green).

between separate occurrences, which is critical for damage quantification and structural health tracking. The few existing instance-level datasets suffer from substantial limitations in terms of scale, resolution, and class diversity. For instance, MCrack1300 [49] provides instance masks for only 1,300 masonry images, while CConCrack [4] offers a limited core set of 400 images for cracks and spalls. Most existing instance datasets focus on a single defect class (predominantly cracks) and lack the large-scale volume of images necessary to train contemporary instance segmentation models effectively.

Table 1 provides a comparative summary of the most influential datasets in this field alongside our proposed work, reporting both native full-scene parameters and derived patch-level metrics. Among classification benchmarks, SDNET2018 [17] and CODEBRIM [38] offer high native resolutions (up to 24 MPx) but rely on low-resolution patches ($\leq$ 0.06 MPx) for primary tasks. In semantic segmentation, CrackSeg9k [28] aggregates over 2,000 images from ten sources into a 0.16 MPx patch benchmark, while dacl10k [19] remains the largest full-scene resource with 9,920 images at 3.10 MPx. Current instance-level efforts like CConCrack [4] (augmented from 400 core images) represent early steps toward defect isolation but are often limited in scale or class diversity. Our CiF dataset addresses these gaps by providing 12,896 high-resolution (16.66 MPx) images with 148,642 high-fidelity tiles (1.05 MPx), representing the first large-scale, multi-class instance segmentation benchmark for bridge inspection that preserves fine-grained detail across both release variants.

3 Dataset Details

3.1 Data acquisition

Collected over five years in collaboration with infrastructure owners (Sund & Bælt, Trafikverket, Zurich Kanton), the dataset focuses on European civil infrastructure, particularly bridges. Its hybrid design combines heterogeneous "in-the-wild" field data to capture real-world variability with controlled inspection setups for structured, repeatable imaging. This dual approach ensures broad structural and environmental diversity, ultimately supporting robust model training and reliable evaluation.

In-the-Wild Data. The in-the-wild subset (12.7% of the data) was collected collaboratively by engineers and researchers during routine inspections of operational infrastructure. Images were

Table 1: Comparison of our dataset parameters with other civil infrastructure defect datasets. For image counts and resolution, we report both the original full-scene data and the derived patches/augmented samples where applicable.

Dataset	Task	Classes	Images		Resolution (MPx)	
			Full	Patches	Full	Patch
SDNET [17]	Classification	1	230	56,092	14.06	0.06
CODEBRIM [38]	Classification	5	1,590	7,733	24.00	0.05
CrackSeg9k [28]	Semantic seg.	1	2,003	9,255	0.25	0.16
S2DS [9]	Semantic seg.	6	743	–	1.05	–
dacl10k [19]	Semantic seg.	19	9,920	–	3.10	–
Özgenel [40]	Instance seg.	1	458	–	12.19	–
MCrack1300 [49]	Instance seg.	5	1,300	–	0.41	–
CConCrack [4]	Instance seg.	2	400	10,995	12.19	1.05
CiF (Ours)	**Instance seg.**	**6**	**12,896**	**148,642**	**16.66**	**1.05**

acquired in unconstrained field conditions using a variety of consumer-grade and industrial camera systems. Consequently, this subset exhibits substantial variability in resolution, viewpoint, illumination, weather conditions, and scene composition. This variability reflects the diversity of real-world inspection scenarios and is intentionally retained to improve model generalisation. In particular, the dataset includes a wide range of defect manifestations across different structural contexts, enabling evaluation under realistic deployment conditions.

Controlled Inspection Setups. Complementing our in-the-wild data, we utilized drone-assisted controlled acquisitions to safely gather scalable, repeatable imagery in difficult environments. To simultaneously maximize structural surface coverage and ensure sufficient pixel density for fine-grained defect detection, we employed two distinct flight modes: a scan setup and a high-resolution setup. Detailed drone and camera specifications are provided in Table 2.

Table 2: Overview of camera systems and lens configurations used in the controlled inspections.

Camera	System type	Focal length [mm]	Zoom / lens	Resolution
DJI ZH20/ZH20T	Optical zoom	6.83–119.94 (opt.)	23× opt., 200× digital	5184 × 3888
DJI Zenmuse X7	Prime lens (S35)	50 (used; 16/24/35/50 avail.)	Fixed lens	6016 × 3376
DJI Zenmuse P1	Prime lens (FF)	50 (used; 24/35/50 avail.)	Fixed lens	8192 × 5460

Scan Setup. Prioritizing efficient, large-scale coverage over maximal detail, the scan setup utilizes a drone flying parallel to the target surface with a perpendicular camera. Operating in downward, vertical, or upward-facing orientations, the drone typically executes a zigzag trajectory to ensure systematic coverage, simultaneously recording high-precision GPS/RTK data to enable the accurate spatial registration of the captured imagery.

High-Resolution Setup. In the high-resolution setup, the drone remains in a stationary hover position while using a gimbal-mounted camera to capture a structured grid of zoomed-in images of the target surface. This approach prioritises fine-grained visual detail. To enable reconstruction of larger surface regions, we assume no oblique views so that homography transformations are used to map local views into a shared reference frame, enabling mosaic reconstruction of the observed area. This setup is particularly suitable for detailed inspection of localized structural regions where high spatial resolution is required for defect inspection. A representative hardware configuration includes the DJI Matrice 300 RTK[2] equipped with a Zenmuse H20T[3], although the proposed acquisition methodology is not restricted to specific hardware and can be implemented with comparable sensor systems.

[2] `https://enterprise.dji.com/matrice-300`

[3] `https://enterprise.dji.com/zenmuse-h20-series/specs`

3.2 Images, Classes and Dataset Splits

The dataset contains 12,896 native high-resolution images (0.23 to 42.18 MPx, mean 16.66 MPx) annotated for six defect categories defined jointly with civil engineers. Unlike most defect benchmarks that release pre-cropped or downsampled patches, we preserve the native resolution: hairline cracks down to a few pixels in width are polygonally annotated within multi-megapixel inspection images, alongside spatially extensive defects such as algae colonisations covering large portions of a single frame. We release two coordinated variants: *Full* preserves the native resolution, while *Tiled* partitions every image into 1024×1024 patches (148,642 in total) using a sliding window with a stride of 768. During tiling, we discard fragmented slivers by requiring that any cropped defect instance retains at least 5% of its total original area, resulting in a uniform-resolution benchmark suitable for current vision backbones.

Classes. The six categories covering the dominant visual defect families on aged concrete and steel surfaces are illustrated in Figure 2. *Algae*: biological surface growth indicating sustained moisture. *Crack*: linear discontinuities in the concrete or coating. *Net-Crack*: interconnected polygonal cracking patterns (alligator or map cracking) associated with fatigue or shrinkage. *Crack with Precipitation*: crack with mineral deposits left by water seepage (efflorescence). *Rust*: oxidation of exposed or embedded steel reinforcement. *Spalling*: detachment or fragmentation of surface material.

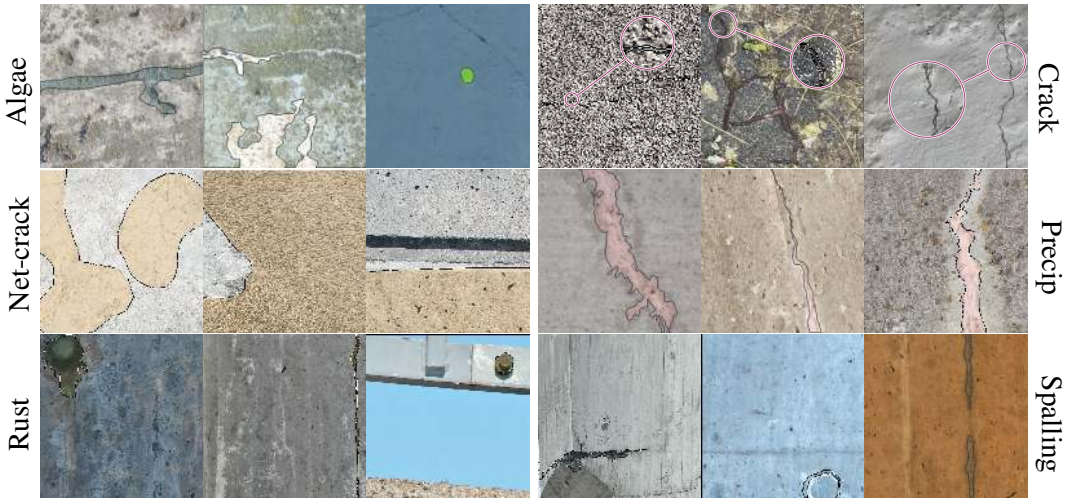


Figure 2: Mosaic of the six defect types in tiled images.

Splits. Images are partitioned into train (70%), validation (15%), and test (15%) at the annotation batch level. Each batch corresponds to an inspection or a defined subset of one, and tiles inherit their parent's split, so no data leaks between splits across resolutions. To prevent spatial leakage, data is organized by owner, asset, and inspection batch; while the same bridge may appear across splits, exact physical locations never overlap because separate inspections target distinct parts of the structure. This methodology yields 12,896 Full images (9,096 train, 1,942 val, 1,858 test) and 148,642 Tiled patches (105,139 train, 22,368 val, 21,135 test), all of which are fully open-sourced alongside the official evaluation splits.

3.3 Annotation Procedure

All defects are annotated as instance-level polygon masks at native image resolution, capturing both hairline cracks a few pixels wide and large-extent defects spanning a substantial fraction of multi-megapixel frames. The class taxonomy of §3 was finalised in close collaboration with civil engineers at Sund & Bælt, restricted to six categories that are simultaneously severe enough to warrant pixel-level localisation and operationally actionable for structural inspection and maintenance. Polygon mask annotation was carried out by an external annotation vendor under written guidelines authored jointly with the domain experts, with quality assurance performed both internally by the vendor and by our in-house reviewers.

Annotation guidelines. A written guideline document was produced for the annotation team, defining each defect type by a textual description, positive examples, and characteristic false positives. Because defect appearance, scale, and confusable background structures vary markedly across acquisition modalities and asset classes, separate guideline variants were issued per image source (drone, phone camera, DSLR) and per structure type (concrete bridge, city infrastructure, etc.). The guidelines were maintained as a living document throughout the five-year campaign: ambiguous samples surfaced by the annotation team were triaged with the domain experts and, once resolved, folded back as new canonical examples. This loop is the primary mechanism by which we controlled the dominant source of label noise in this domain—defect ambiguity at the boundary between visually similar categories. Our segmentation masks strictly adhere to standard inspection practices [3, 11]. Net-Crack and Spalling are area-based defects, so we annotate them as filled blobs. Individual Cracks are linear, so we annotate them as thin polygons and derive their width post-hoc.

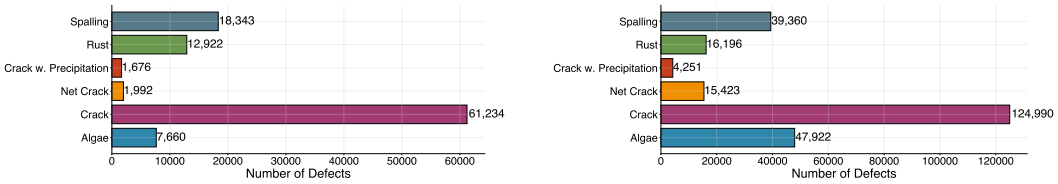
Pre-annotation triage. Two classes of images were excluded prior to annotation. Frames judged to be of insufficient image quality (motion blur, severe defocus, or extreme exposure) were discarded. Frames that contained no defect of interest were treated as background and excluded from the annotated set; they do not contribute to the released splits.

Batches and pilot phase. Annotation proceeded in batches of approximately 500–1,000 images, grouped by image source and structure type, which we refer to as a *batch type*. For every new batch type, the annotation team first delivered a pilot of 10–50 images, used both to calibrate annotators to the specific imaging conditions and structure and to surface guideline gaps before they propagated through a full batch. Pilots were reviewed end-to-end and iterated upon with the team until quality was acceptable; a full batch was released only after pilot sign-off.

Full-batch annotation and QA. Full annotation batches were produced by an external vendor, who conducted an initial internal Quality Assurance (QA) pass prior to delivery. Upon receipt, our in-house ML research team—trained by domain experts, primarily at Sund & Bælt—executed a rigorous second-pass audit by performing an instance-level review on a representative 10% sample of the images. If this audit revealed an instance-level false-positive or false-negative rate exceeding our strict 10% threshold, the entire batch was rejected and returned for correction; otherwise, localized errors were rectified and the batch was approved. To establish a robust baseline, the first five batches underwent a comprehensive 100% review. This initial bootstrapping phase served both to calibrate our internal reviewers against the annotation guidelines and to empirically validate that our 10% sampling protocol accurately captured overall batch quality. Comprehensive details of our QA pipeline are provided in Appendix A.5.

3.4 Statistics

The *Full* variant is characterized by its high native resolution, which significantly exceeds that of standard generic instance segmentation benchmarks. With a mean of 16.66 MPx and a median of 20.16 MPx, the images are approximately 50× larger than COCO (∼0.3 MPx) [33] and 8× larger than Cityscapes (2.1 MPx) [15]. The cumulative distribution shown in Fig. 4b reveals that approximately 70% of the images have a resolution of at least 20 MPx, reflecting the raw output of operational field hardware: smartphones, DSLRs, and gimbal-mounted survey cameras. Such high resolution is essential to capture hairline defects and high-frequency structural textures that would be lost in standard downsampled benchmarks.



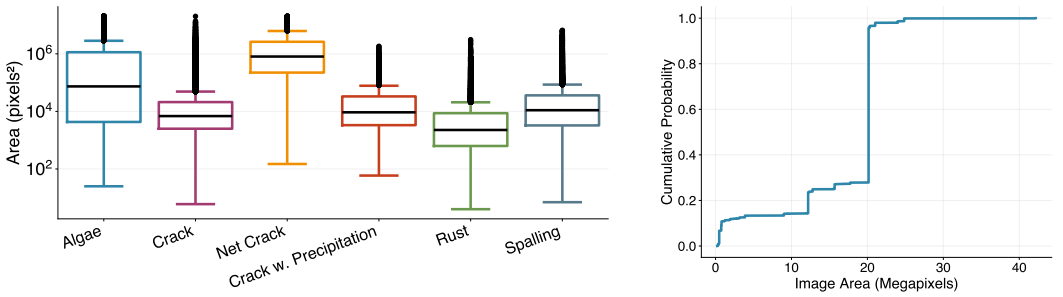
(a) Full variant.

(b) Tiled variant.

Figure 3: Per-class defect instance counts in the Full and Tiled variants.

Instance Density and Class Distribution. The dataset contains a total of 103,827 defect instances in the Full variant. On average, each image contains 8.05 instances, a density that is higher than both generic benchmarks like COCO [33] (∼7.3 instances/image) and specialized infrastructure benchmarks like CODEBRIM [38] (∼3.37 instances/image). The class distribution is strongly long-tailed (Fig. 3). In the Full variant, Crack accounts for 59.0% of all instances, while Spalling and Rust represent 17.7% and 12.4% respectively. Rarer categories such as Net-Crack (1.9%) and Crack with Precipitation (1.6%) still provide nearly two thousand instances each (Net-Crack: 1,992, Precipitation: 1,676 in Full). The Tiled variant exhibits a different distribution due to the partitioning of spatially extensive defects: Algae rises to 19.3% of instances, while Crack and Spalling account for 50.4% and 15.9% respectively. We preserve the natural frequency of defect classes without artificially resampling or balancing the data. This ensures the dataset's statistical prior accurately reflects real-world conditions, where cracks naturally dominate as the primary physical precursor to secondary deterioration such as rust and spalling [24].

Defect Scale and Geometric Diversity. Beyond image scale, the difficulty profile is dominated by the wide span of defect instance areas across and within classes. Median instance areas span over two orders of magnitude across categories: from ∼2.3 K px^2 for Rust to ∼807 K px^2 for Net-Crack. Intra-class spread is exceptionally large, with Crack instances ranging over six orders of magnitude in area (Fig. 4a). While the Full variant preserves these extremes, the Tiled variant filters out small defects (minimum area ∼0.5 K px^2) and partitions large defects (e.g., Algae and Net-Crack) into many tiles. Furthermore, the defects exhibit high geometric diversity. While Cracks are characteristically elongated (mean aspect ratio 4.4), Algae and Net-Cracks often cover irregular, spatially extensive areas with higher mean compactness (0.67 and 0.85, respectively, in Full). This structural variety prevents models from relying on simple shape priors and requires robust spatial reasoning across multiple scales.



(a) Defect instance area per class.

(b) Cumulative image resolution distribution.

Figure 4: Quantitative properties of the Full variant: distribution of defect areas across classes (a) and cumulative native image resolution distribution (b).

Baselines in this paper operate on the Tiled variant; additional experiments on the Full variant are reported in Appendix A.4; the Full variant is released to support evaluation of resolution-aware and tile-aggregation methods. Class co-occurrence and per-class shape compactness are reported in Appendices A.2, and A.3.

4 Evaluation

Supervised baselines. To establish rigorous baselines on the CiF dataset, we evaluate a diverse suite of state-of-the-art architectures renowned for top-tier performance on standard instance segmentation benchmarks like COCO. Our selection strategically covers three distinct paradigms: (i) Industry Standards: We include the YOLO family (specifically YOLOv11 [26] and YOLOv26 [46]) as the de facto standards for real-world, high-performance segmentation; (ii) DETR-Based Architectures: We evaluate Mask DINO [31] as a strong DETR-based [13] segmentation and detection baseline, alongside RF-DETR [44], a state-of-the-art compact and computationally efficient transformer model; and (iii) Pretrained Foundations: We include EoMT [25] to assess knowledge transfer of strong, pretrained foundational encoders to the civil infrastructure domain. Specifically, we aim to probe the

Table 3: Object detection (bb) and instance segmentation (seg) results for supervised baselines on the CiF dataset. Models were trained on the training set, and the best-performing checkpoint on the validation set was used for final evaluation on the test set.

Method	Encoder	$mAP_{50:95}$		$AP_{50:95}$											
				Algae		Crack		Net-Crack		Precipitation		Rust		Spalling	
		bb	seg	bb	seg	bb	seg	bb	seg	bb	seg	bb	seg	bb	seg
Mask-DINO [31]	Resnet-50	29.0	20.4	51.2	46.5	31.3	3.0	27.2	22.8	14.1	4.3	26.9	26.7	23.4	19.0
YOLOv11 [26]	nano	31.0	19.8	52.6	43.6	31.0	6.5	29.0	21.0	17.1	4.9	30.0	25.1	26.1	17.7
	small	32.3	20.7	53.0	43.9	32.9	7.5	29.2	21.7	18.3	5.9	32.8	26.7	27.4	18.7
	medium	33.2	21.5	53.4	44.3	34.6	8.2	30.2	23.2	18.8	6.3	34.0	27.4	28.4	19.7
	large	33.9	21.9	53.8	44.7	35.6	8.5	30.8	23.5	19.9	6.7	34.7	27.8	29.2	20.3
	xlarge	33.9	21.9	53.6	44.3	35.6	8.6	30.7	23.4	19.7	7.2	35.0	28.2	28.7	19.9
YOLOv26 [46]	nano	30.2	19.5	52.0	43.5	30.3	6.8	27.8	20.7	17.4	4.9	28.2	23.6	25.2	17.4
	small	32.2	20.9	52.7	43.9	32.5	7.6	29.1	21.8	19.6	7.0	31.7	25.9	27.6	19.0
	medium	32.8	21.4	53.0	44.3	33.2	8.1	29.8	22.9	20.2	6.9	33.3	27.3	27.3	19.1
	large	32.0	20.2	51.7	42.8	33.1	7.9	27.5	20.4	20.1	6.3	32.4	25.6	26.8	18.5
	xlarge	32.7	21.0	52.2	43.4	34.1	**8.9**	27.2	20.6	20.1	6.8	33.7	26.2	28.9	20.3
RF-DETR [44]	nano	35.3	18.3	**59.3**	46.1	37.5	1.2	30.6	21.5	18.6	7.3	36.4	16.7	29.7	17.1
	small	**36.2**	18.2	59.2	46.2	**38.3**	1.3	**31.8**	21.4	**20.5**	7.2	**36.8**	17.4	**30.5**	15.8
	medium	35.7	18.3	59.1	46.8	38.1	1.2	31.5	21.5	19.6	6.8	36.4	17.0	29.7	16.7
	large	35.7	18.3	58.9	46.5	38.2	1.3	31.5	21.0	19.5	7.6	36.5	16.7	29.8	16.6
EoMT [25]	DinoV3	32.9	**24.2**	57.0	**49.6**	28.4	4.7	30.4	**24.1**	17.9	**10.3**	34.8	31.6	29.2	25.1
	DinoV2	32.7	24.1	56.4	48.8	29.0	5.0	29.3	23.5	16.8	9.5	35.2	**32.6**	29.6	**25.2**

out-of-domain finetuning of foundation backbones like DINOv2/v3 [39, 48] using a state-of-the-art algorithm like EoMT. All baselines are fully fine-tuned on the CiF training split, starting from standard pretrained weights if available with optimal checkpoints selected via validation performance before final evaluation on the held-out test set. The results are detailed in Table 3, reporting the $\text{mAP}_{50:95}$ performance for both the instance segmentation masks and the bounding box predictions. More details about the training setups can be found in Appendix A.1.

Foundation Model Evaluation. Given the recent success of Foundation Models (FMs) and Vision-Language Models (VLMs) claiming universal, zero-shot detection and segmentation capabilities [27, 30, 7], we conduct a rigorous evaluation of these architectures on the CiF dataset. We specifically investigate two dominant paradigms: promptable segmentation and zero-shot object detection. For promptable segmentation, we evaluate SAM3 [12], the latest iteration of the Segment Anything family [27, 43]. To thoroughly assess its adaptability, we evaluate SAM3 under three distinct prompting protocols: oracle bounding boxes derived from ground truth, oracle point-based prompting (randomly sampling 1 to 5 points from the ground truth mask), and text-based prompting. Because these oracle protocols rely on privileged ground-truth spatial prompts, they are designed strictly to evaluate SAM3's viability as a semi-automated annotation assistant rather than as an off-the-shelf segmenter. For zero-shot detection, we evaluate Qwen3-VL, a state-of-the-art VLM known for exceptional performance on referring expression segmentation benchmarks (e.g., refCOCO, refCOCO+ [51]) and diverse vision-language tasks. Qwen3-VL is evaluated using standard off-the-shelf text prompts, instructed to output object detections in a structured JSON format (Prompt details can be found in Appendix A.6). Moreover, we benchmark SAM3-Agent, a hybrid architecture integrating the zero-shot reasoning capabilities of Qwen3-VL with the precise dense prediction of SAM3 within an agentic framework. Quantitative results are detailed in Table 4 for instance segmentation and bounding box detection, with both evaluated using the $\text{mAP}_{50:95}$ metric.

4.1 Discussion

Moderate success of supervised baselines. Standard supervised models achieve reasonable object detection performance, yielding an overall bounding-box mAP@50:95 of approximately 35%. The models exhibit strong localization capabilities on visually distinct, surface-level anomalies such as algae (reaching up to 59% mAP). However, performance degrades on critical structural defects. Complex failure modes, such as *net crack* or *crack with precipitation*, yield localization scores of only 20–30%, showing all the challenges of the civil infrastructure domain. When it comes to

Table 4: Performance comparison of SAM3 and Qwen3-VL on the CiF dataset. The $AP_{50:95}$ metric denotes segmentation accuracy for SAM3 and object detection accuracy for Qwen3-VL. Note that SAM3-Oracle variants utilize privileged ground-truth spatial prompts strictly to assess semi-automated labeling viability and are not directly comparable to unprivileged zero-shot baselines (SAM3-Text, SAM3-Agent, and Qwen3-VL).

Method	$mAP_{50:95}$	$AP_{50:95}$					
		Algae	Crack	Net-Crack	Precipitation	Rust	Spalling
SAM3-Agent	1.7	1.0	0.7	0.0	1.0	7.0	0.7
SAM3-Text	2.0	1.0	0.7	0.0	1.6	8.8	0.0
SAM3-Oracle #1 point	7.8	10.4	0.4	6.7	3.0	18.1	8.2
SAM3-Oracle #3 points	16.1	30.5	0.6	20.9	4.6	25.5	14.7
SAM3-Oracle #5 points	18.6	33.9	0.8	25.0	5.8	28.2	17.9
SAM3-Oracle Bounding Box	34.2	56.0	1.9	54.1	10.4	46.0	37.1
Qwen3-VL-4B	1.5	4.7	2.4	0.1	0.1	1.4	0.5
Qwen3-VL-8B	1.9	5.8	2.0	0.1	0.1	2.8	0.4
Qwen3-VL-32B	2.0	6.7	2.8	0.4	0.1	1.3	0.6
Qwen3-VL-235B-A22B	2.4	7.3	3.3	0.7	0.1	2.3	0.8

instance segmentation, performance drops drastically across all architectures. With the exception of high-contrast algae, mask mAP plunges to single digits for fine-grained structural defects like cracks. This drop in pixel-level accuracy is particularly detrimental for civil engineering applications, where the severity of a defect is strictly measured by its exact geometric footprint. Among the segmentation baselines, EoMT achieves the highest performance by leveraging an internet-scale foundational backbone. This outcome suggests that massive-scale pretraining may offer a measurable advantage in feature extraction, though it is difficult to cleanly disentangle the impact of the foundational data from the model's inherent architecture. What remains definitively clear, however, is that the absolute performance is insufficient for real-world application. This stark reality positions CiF as a vital asset and proving ground, challenging the field to determine whether future breakthroughs in civil infrastructure will be driven by specialized architectural advancements or by even larger, high-fidelity datasets.

The Illusion of Zero-Shot Generalization. Our evaluation of Foundation Models and VLMs reveals poor zero-shot performance, exposing a critical gap between benchmark success and real-world utility. For instance, despite its massive scale, the Qwen3-VL (235B-A22B) model fundamentally fails at zero-shot object detection on civil infrastructure, yielding an abysmal mean bounding box mAP of just 2.4%. SAM3 fares no better in zero-shot dense prediction. Even when evaluated under oracle conditions (using ground-truth bounding boxes or point prompts), performance remains strikingly low except for large defects. The model struggles with the fine-grained predictions required for the "crack" class, confirming that generic segmentation foundations lack the specific priors necessary for defect segmentation. Consequently, these limitations undermine SAM3's viability as an automated annotation assistant, further reinforcing the extreme difficulty and expense of curating high-fidelity training data in this domain.

Cracks in the Foundation. The evaluation experiments point to an overlooked, yet vital issue in current ML literature: models that achieve state-of-the-art results on curated, general-purpose datasets consistently fail to generalize to complex, real-world domains such as civil infrastructure visual inspection. The primary objective of CiF is to spotlight this deficiency and open-source a massive-scale dataset to close this gap. Given the profound societal and economic impact of automated structural health monitoring, resolving this domain shift is not just a theoretical machine learning challenge, but a critical real-world necessity.

4.2 Limitations

While CiF represents a large-scale and relevant dataset for civil infrastructure defect segmentation, several limitations remain. First, the dataset focuses primarily on bridge structures and concrete materials in Europe, and therefore does not fully represent the diversity of global infrastructure

systems, such as steel or underground structures. Second, defect annotation is inherently subjective, particularly for fine-grained or low-contrast damage patterns, which may introduce variability even among expert annotators. Third, the data acquisition process—combining in-the-wild imagery with controlled drone-based inspections—may introduce biases in viewpoint, scale, and sensor characteristics. In addition, the dataset reflects a long-tailed distribution of defect types, which may impact model performance across rare categories. Lastly, despite its unprecedented scale, CiF remains vastly smaller than general-purpose datasets like LAION [47]. While our evaluation exposes severe model failures, the root cause remains an open question. We cannot prove whether these limitations are strictly architectural; it is possible that scaling the dataset by several orders of magnitude could resolve them, though such an effort is currently economically prohibitive.

5 Conclusions

We introduce Cracks in the Foundation (CiF), the largest high-resolution instance segmentation dataset for civil infrastructure to date. Evaluations on CiF suggest the domain remains highly challenging: state-of-the-art Foundation Models struggle with zero-shot generalization, and specialized models plateau below real-world requirements. Ultimately, CiF establishes structural health monitoring as an open, unsolved challenge. By open-sourcing this unprecedented benchmark and large-scale training dataset, we empower the community to develop robust architectures capable of genuinely safeguarding our physical world.

Acknowledgments and Disclosure of Funding

We would like to sincerely thank Finn Bormlund and Svend Gjerding (Sund & Baelt), Jens Häggström (Trafikverket), Raphael von Thiessen (Innovation-Sandbox for AI, Office for Economy, Kanton Zürich), and the Dübendorf Air Base for granting us the opportunity to collect, analyze, and disseminate the images and defect data included in this publication. We acknowledge funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie actions HORIZON-MSCA-2022-DN-01 call (Grant agreement ID: 101119554) to Nicola Farronato and Rizwan Ullah Khan, and Horizon Europe grant No. 101070408 (SustainML) to Niccolo Avogaro and Mattia Rigotti.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Palisa Arafin, AHM Muntasir Billah, and Anas Issa. Deep learning-based concrete defects classification and detection using semantic segmentation. *Structural Health Monitoring*, 23(1):383–409, 2024.

[3] ASTM International. Standard practice for roads and parking lots pavement condition index surveys. Standard D6433-24, ASTM International, 2024.

[4] Saeid Ataei, Saeed Adibnazari, and Seyyed Taghi Ataei. Data-driven detection and evaluation of damages in concrete structures: Using deep learning and computer vision. *arXiv preprint arXiv:2501.11836*, 2025.

[5] Niccolò Avogaro, Thomas Frick, Mattia Rigotti, Andrea Bartezzaghi, Filip Janicki, A. Cristiano I. Malossi, Konrad Schindler, and Roy Assaf. Show or tell? effectively prompting vision-language models for semantic segmentation. *Transactions on Machine Learning Research*, 2025.

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966, 2023.

[7] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[8] Christian Benz, Paul Debus, Huy Khanh Ha, and Volker Rodehorst. Crack segmentation on uas-based imagery using transfer learning. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2019.

[9] Christian Benz and Volker Rodehorst. Image-based detection of structural defects using hierarchical multi-scale attention. In *DAGM German Conference on Pattern Recognition*, pages 337–353. Springer, 2022.

[10] Eric Bianchi and Matthew Hebdon. Visual structural inspection datasets. *Automation in construction*, 139:104299, 2022.

[11] Bundesministerium für Verkehr und digitale Infrastruktur. *Richtlinie zur einheitlichen Erfassung, Bewertung, Aufzeichnung und Auswertung von Ergebnissen der Bauwerksprüfungen nach DIN 1076 (RI-EBW-PRÜF)*, 2017. Stand: 22.02.2017.

[12] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pages 138846–138923, 2026.

[13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[14] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28652–28668, 2026.

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[16] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 91–104. IEEE, 2025.

[17] Sattar Dorafshan, Robert J Thomas, and Marc Maguire. Sdnet2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in brief*, 21:1664–1668, 2018.

[18] Charles R Farrar and Keith Worden. *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[19] Johannes Flotzinger, Philipp J Rösch, and Thomas Braml. dacl10k: benchmark for semantic bridge damage segmentation. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8611–8620. IEEE, 2024.

[20] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[21] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5351–5359. IEEE, 2019.

[22] Philipp Huethwohl. Cambridge bridge inspection dataset. Apollo, University of Cambridge Repository, 2017. `https://doi.org/10.17863/CAM.13813`.

[23] Philipp Hüthwohl, Ruodan Lu, and Ioannis Brilakis. Multi-classifier for reinforced concrete bridge defects. *Automation in Construction*, 105:102824, 2019.

[24] Fragkoulis Kanavaris, Mário Coelho, Nuno Ferreira, Miguel Azenha, and Carmen Andrade. A review on the effects of cracking and crack width on corrosion of reinforcement in concrete. *Structural Concrete*, 24(6):7272–7294, 2023.

[25] Tommie Kerssies, Niccolo Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan De Geus. Your vit is secretly an image segmentation model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25303–25313. IEEE, 2025.

[26] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pages 3992–4003. IEEE, 2023.

[28] Shreyas Kulkarni, Shreyas Singh, Dhananjay Balakrishnan, Siddharth Sharma, Saipraneeth Devunuri, and Sai Chowdeswara Rao Korlapati. Crackseg9k: A collection and benchmark for crack segmentation datasets and frameworks. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 179–195, Cham, 2023. Springer Nature Switzerland.

[29] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, page 611–626, 2023.

[30] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589. IEEE, 2024.

[31] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050. IEEE, 2023.

[32] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. arXiv preprint arXiv:2407.07895, 2024.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[35] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[36] Arman Malekloo, Ekin Ozer, Mohammad AlHamaydeh, and Mark Girolami. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, 21(4):1906–1955, 2022.

[37] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, et al. Mm1: methods, analysis and insights from multimodal llm pre-training. In *European Conference on Computer Vision*, pages 304–323. Springer, 2024.

[38] Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 11188–11197. IEEE, 2019.

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[40] Çağlar Fırat Özgenel. Concrete crack segmentation dataset. Mendeley Data, V1, 2019.

[41] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824, 2023.

[42] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13009–13018. IEEE, 2024.

[43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pages 28085–28128, 2025.

[44] Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, volume 2026, pages 56087–56100, 2026.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[46] Ranjan Sapkota, Rahul Harsha Cheppally, Ajay Sharda, and Manoj Karkee. Yolo26: key architectural enhancements and performance benchmarking for real-time object detection. *arXiv preprint arXiv:2509.25164*, 2025.

[47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[48] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. Featured Certification.

[49] Zehao Ye, Lucy Lovell, Asaad Faramarzi, and Jelena Ninić. Sam-based instance segmentation models for the automation of structural damage detection. *Advanced Engineering Informatics*, 62:102826, 2024.

[50] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations*, volume 2024, pages 57153–57180, 2024.

[51] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pages 69–85. Springer, 2016.

[52] Zhaohui Zheng, Yuming Chen, Qibin Hou, Xiang Li, Ping Wang, and Ming-Ming Cheng. Zone evaluation: Revealing spatial bias in object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8636–8651, 2024.

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5122–5130. IEEE, 2017.

A Technical Appendices and Supplementary Material

A.1 Training Details

All baselines were trained using the off-the-shelf configurations provided in their respective official implementations. The single exception was the YOLOv26-large and YOLOv26-xlarge models, where the default learning rate caused training divergence; consequently, we scaled the learning rate down by an order of magnitude. We consciously refrained from exhaustive hyperparameter optimization. While extensive tuning might yield marginal performance gains, it incurs prohibitive computational costs and unnecessary environmental impact. Furthermore, aggressive tuning falls outside the scope of this work: the primary objective of CiF is to establish global baseline trends and release a large-scale foundational dataset, rather than to overfit specific architectures to the domain.

Due to the scale of the dataset and the number of models evaluated, the experiments required substantial computational resources. All experiments were conducted on the IBM Research internal cluster. The main training phase for the 17 baseline models utilized 17 compute nodes, each equipped with 8 NVIDIA A100 GPUs, for approximately 1 day. Including preliminary experiments, and validation runs, the total project consumed approximately 10,000 NVIDIA A100 GPU hours. All evaluation runs were performed on a single node with 8 NVIDIA A100 GPUs, requiring approximately 2 days.

A.2 Class Co-occurrence

Table 5: Per-class median compactness on the Full variant.

Class	Compactness (median)
Algae	0.69
Crack	0.58
Net-Crack	1.00
Precipitation	0.29
Rust	0.68
Spalling	1.00

Figure 5 reports the frequency with which pairs of defect categories appear in the same image (Full variant). Crack and Spalling co-occur most frequently (3,273 images), followed by Rust and Spalling (2,201) and Crack and Rust (2,185), illustrating the presence of multiple defect types within individual scenes.

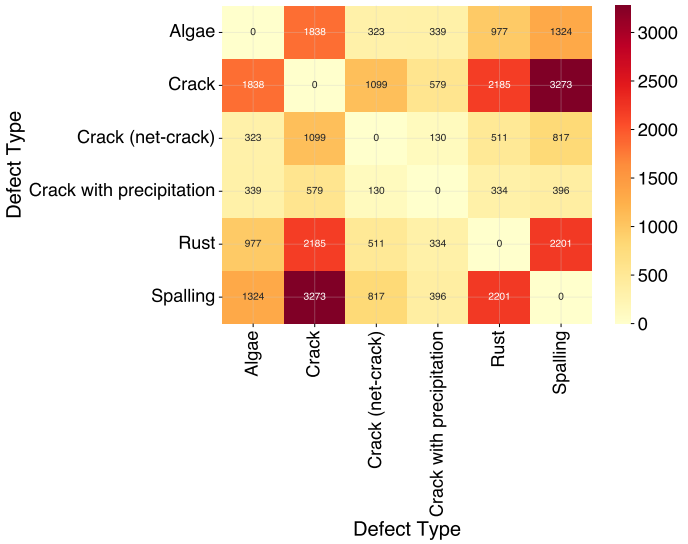


Figure 5: Co-occurrence frequency between defect categories within the same image (Full variant).

A.3 Per-class Shape Compactness

Compactness is the ratio of the instance area to the area of its convex hull, taking values in $[0, 1]$ with values close to 1 indicating convex, blob-like shapes. Table 5 reports per-class median values on the Full variant. Net-Crack and Spalling are the most compact categories (median 1.00), consistent with their tendency to form simple, convex geometries or polygonal networks. In contrast, Crack with Precipitation has the most irregular boundaries (0.29), driven by diffuse deposition patterns. Crack sits slightly above the midpoint (0.58), reflecting the irregular, branching geometry of individual fractures.

A.4 Full Resolution Experiments

Our images average 16.66 Megapixels and reach up to 42.18 Megapixels. We initially attempted to process them natively, but because a single full-resolution sample causes out-of-memory errors even on NVIDIA A100 80GB GPUs, we trained our "full resolution" baselines using the maximum downscaled input size that would fit into memory on a single A100. These effective dimensions are reported in the "Resolution" column of Table 6. We can see the general performance drop: as anticipated, the metrics are exceptionally poor across the entire evaluation when models are forced to handle these massive resolutions. Between the evaluated models, YOLO models perform slightly better in this regime, most likely because their architecture is inherently more scalable to high-resolution inputs.

Table 6: Results on the Full variant of CiF. The $AP_{50:95}$ metric denotes instance segmentation accuracy for the first three supervised rows and SAM3, and object detection accuracy for the remaining models. Input dimensions are listed in the Resolution column.

Method	Resolution	$mAP_{50:95}$	$AP_{50:95}$					
			Algae	Crack	Net-Crack	Precipitation	Rust	Spalling
EoMT (Segmentation)	1728×2304	5.6	15.4	0.5	5.0	0.9	7.6	4.6
YOLOv11-m	2388^2	7.1	2.8	4.2	0.8	2.0	19.1	13.6
YOLOv26-m	2388^2	10.6	15.2	6.3	3.4	2.3	20.8	15.3
RF-DETR (Detection)	4192^2	7.7	10.5	4.7	1.1	0.7	18.7	10.7
YOLOv11-m	2388^2	10.9	3.6	12.1	1.3	5.0	24.1	19.2
YOLOv26-m	2388^2	16.1	19.4	16.7	5.5	7.0	26.7	21.3
SAM3-Agent	1024^2	0.4	0.0	0.0	0.0	0.0	2.6	0.0
SAM3-Text	1024^2	1.2	0.0	0.0	0.0	0.1	6.5	0.0
SAM3-Oracle #1 point	1024^2	3.8	6.3	0.0	0.5	0.7	9.9	5.5
SAM3-Oracle #3 points	1024^2	4.1	8.0	0.0	2.7	0.9	7.7	5.5
SAM3-Oracle #5 points	1024^2	4.3	8.8	0.0	3.3	2.2	6.9	4.8
SAM3-Oracle Bounding Box	1024^2	21.2	34.3	0.2	31.2	7.0	29.4	24.9
Qwen3-VL-4B	full-resolution	0.7	1.9	0.6	0.0	0.1	1.3	0.4
Qwen3-VL-8B	full-resolution	0.6	0.9	0.5	0.1	0.3	1.1	0.8
Qwen3-VL-32B	full-resolution	0.6	1.7	0.6	0.0	0.2	0.6	0.3

A.5 Annotation Quality Control

Limitations of standard multi-annotation: Standard dataset creation often relies on consensus across multiple independent annotators [21], but we found this conventional approach unworkable for infrastructure defects. Early in the project, we had multiple non-expert annotators independently label a subset of images, which resulted in notably low agreement (25% recognition quality at 68% segmentation quality). Crucially, this divergence was driven by disagreements over whether a defect was actually present, rather than how to draw its boundaries. Identifying structural anomalies requires specific domain expertise that a simple majority vote among standard annotators cannot replace. In light of this fundamental limitation, we designed a custom expert-review process for our needs, which sidesteps the pitfalls of independent consensus to reliably reduce residual errors.

Our QA procedure: An external vendor labeled each batch according to extensive annotation guidelines and conducted a first-pass quality assurance (QA) process. Our in-house team, trained by domain experts, then performed a second-pass review on a 10% image sample per batch, scoring instance-level false positives (FP) and false negatives (FN). Batches above 10% FP/FN were returned

to the vendor for full re-annotation/correction and re-reviewed afterwards; batches below were accepted after correcting the found errors. Blurry, defocused, or badly exposed frames were discarded before annotation, so image blur is not a source of label error in the released dataset. Across all batches, our QA procedure produced 47,795 instance-level corrections over 2,922 re-annotated/corrected images.

(1) Proportion of expert review: In-house annotators trained by the domain experts reviewed a 10% image sample per batch, which then served as the accept/reject trigger. Failing batches were fully re-annotated/corrected and re-reviewed, so the effective expert-touched fraction is substantially higher for the batches that needed it.

(2) Accepted vs. rejected batches: Of a total of 15 batches, eight were rejected on first review, with first pass error rates of up to 49% FP and 42% FN. They were then re-annotated and re-reviewed before acceptance. The rest were accepted after small corrections. Every released batch met the $\leq 10\%$ criterion at acceptance. The high first-pass error is precisely what the review loop is designed to catch, and none of it is present in the released data.

(3) Realized error at acceptance: Accepted batches have a final error rate of about 3% FP and 4% FN on average, well below our strict 10% threshold. We calculate these final numbers based on the last audit step right before a batch is approved. During this stage, our experts only correct the mistakes they find inside the 10% audit sample. To estimate the remaining error, we project the mistake rate found in this sample onto the unreviewed 90% of the batch. This projection assumes the 10% sample accurately represents the whole batch. Naturally, these rates vary depending on the batch and the specific defect. Cleanly annotated batches with well-defined categories have error rates under 1%. In contrast, the most visually ambiguous cases, such as easily missed rust or algae, can reach up to a 10% FN rate.

(4) Per-class boundary error. To characterize boundary quality per class, we compare each released label against its pre-correction version on the re-annotated batches, using mask IoU and Boundary IoU (overlap within a 3 px band around the contour, which penalizes edge shifts that mask IoU hides). The results are presented in Table 7 where agreement is high for compact, well-defined classes (crack, spalling, rust; Boundary IoU ≥ 0.97) and lowest for the wide-area or visually ambiguous classes (net-crack 0.79, algae 0.91), identifying these as the hardest to delineate.

Table 7: Per-class mask IoU and Boundary IoU between released labels and their pre-correction versions on re-annotated batches.

Class	Mask IoU	Boundary IoU (3px)
Crack	0.975	0.974
Crack with Precipitation	0.972	0.971
Spalling	0.988	0.976
Rust	0.991	0.978
Algae	0.969	0.908
Net-Crack	0.918	0.786

(5) Ambiguous samples. Class relabeling during review was rare: nearly all corrections were boundary refinements or missed instances rather than reassignments between classes. Ambiguity therefore surfaces as boundary disagreement, concentrated in the classes identified above. The residual ambiguous cases are presence decisions rather than extent decisions, principally whether a dark linear feature is a crack or a shadow.

A.6 Zero-Shot and Foundation Model Evaluation Details

To evaluate out-of-the-box generalization rather than prompt-engineering sensitivity, all foundation models were tested using their official, minimally modified prompting protocols.

Qwen3-VL. Inference was conducted via vLLM [29] using greedy decoding to ensure deterministic outputs. Following the official Qwen3-VL 2D grounding cookbook, we queried the models with the standard prompt: *"Locate every instance that belongs to the following categories: 'algae, crack, net-crack, crack with precipitation, rust, spalling'. Report bbox coordinates in JSON format."* Outputs were processed using the official Qwen3-VL parsing script. Formatting and parsing errors occurred in

a negligible 0.3% of test samples, confirming that the low detection performance reflects fundamental visual grounding limitations rather than output-parsing failures.

SAM3. For SAM3-Text, the model was prompted directly with the target class name. For SAM3-Agent, we retained the strictly unmodified system prompts governing its tool-use protocol (e.g., `segment_phrase`, `examine_each_mask`) and iterative mask-verification sub-loop. Domain adaptation was restricted entirely to the user query: *"the model should produce a mask for each {category} in this image of a civil infrastructure defect survey."*

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim a dataset contribution, and we made the entire dataset publicly available. Dataset details (see Section 3), sources (see Subsection 3.1), and annotation protocols (see Subsection 3.3) are fully documented. As a reference, we have fully fine-tuned a family of supervised models (see Table 3), and we also evaluate SAM3 and Qwen3-VL (see Table 4) to demonstrate the challenge the dataset poses to established models.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we are fully aware and attribute limitations in Subsection 4.2.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This work provides a dataset accompanied by empirical evaluations.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose the information needed to reproduce the experimental results. The dataset splits and evaluation metrics are described in Section 3 and Section 4. For the baseline evaluations, we utilized the models directly from their official repositories, applying the default hyperparameters and training procedures as specified in their respective original papers, when not differently specified.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While the dataset itself is made fully open access with instructions for downloading and usage, we do not provide a standalone code repository for the baseline experiments. This is because we strictly used the unmodified original codebases of the respective models to test the dataset, adopting the hyperparameters detailed in their original publications when not specified.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant data split details are provided in Section 3. As we utilized existing models from their original repositories to benchmark the dataset, the training details, hyperparameters, and optimizer choices remain identical to those reported in the original authors' papers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The core contribution of this work is the release of the dataset. Due to the substantial computational requirements of fully training the AI models, we did not repeat those experiments. However, note that our main claim - providing a challenging dataset - remains valid even if the reported performance numbers fluctuate significantly. Therefore, we present a broad overview of model performance rather than making specific claims about which model performs best.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the compute resources required in Appendix A.1. Briefly, training the 17 main baseline models utilized 17 nodes with 8 NVIDIA A100 GPUs each. The total project compute, including preliminary runs, is estimated at 10,000 GPU hours on the IBM Research cluster.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that this research complies with applicable ethics guidelines. In particular, all work carried out by third parties (including access to infrastructure, data collection, and annotation) is governed by contracts that undergo strict internal ethical and

legal review procedures. Data-related considerations have been addressed to ensure that we have the right to release the dataset, as all data was either captured or produced by our partners or directly by us. We have also ensured that the released data is not sensitive, as it contains no personal information and does not include images in which individual human subjects are visible.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both the potential positive and negative societal impacts of releasing the CiF dataset. On the positive side, improving automated structural health monitoring can contribute to safer infrastructure by enabling earlier and more accurate detection of defects, potentially reducing the risk of catastrophic failures. Additionally, the dataset may accelerate research in dense image understanding and lower barriers to entry for developing and evaluating models in this domain.

On the negative side, we acknowledge potential risks, including misuse of the dataset (e.g., over-reliance on imperfect models in safety-critical settings), biases or limitations in the data that could lead to unreliable performance in underrepresented scenarios, and the possibility that poor model generalization may give a false sense of reliability. We therefore emphasize that the dataset is intended for research purposes and highlight the need for careful validation before deployment in real-world safety-critical applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released dataset does not fall into a high-risk category for misuse, as it contains only images of civil infrastructure and does not include personal data or sensitive content. Nevertheless, we take responsible release seriously. All data has been collected and curated under strict ethical and legal review procedures, and we ensure that we have the right to release it. We also emphasize that the dataset is intended for research purposes and caution against deploying models trained on it in safety-critical applications without thorough validation.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset presented in this work is either originally created by the authors or by collaborating partners, and appropriate credit is given where applicable. Data collection, annotation, and infrastructure usage are governed by formal agreements that ensure compliance with legal and ethical standards. We confirm that we have the rights to use and release the dataset, and that all relevant licenses and terms of use are respected. Where external code or models are used for evaluation, we follow their respective licenses and provide proper attribution.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the CiF dataset and fully document it in the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Details of the annotation process and a summary of the instructions given to the vendors are provided in Section 3.3. All acquisition and annotation tasks were performed by professional partners and an external vendor under contracts that ensure fair compensation and labor practices in accordance with the NeurIPS Code of Ethics.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Since this research does not involve human subjects, Institutional Review Board (IRB) approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

Justification: Large Language Models were used strictly for writing, formatting, and editing assistance, and do not constitute an important, original, or non-standard component of the core research methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.