International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Computer Vision-Based Automated Crack Detection and Severity Classification in Reinforced Concrete Structures

VIJAY SAINI
Assistant Professor, Department of Civil Engineering
Jagannath University, Jaipur, Rajasthan, India

https://doi.org/10.55041/ijsmt.v2i11.028

Cite this Article: SAINI, V. (2026). Computer Vision-Based Automated Crack Detection and Severity Classification in Reinforced Concrete Structures. International Journal of Science, Strategic Management and Technology, *02*(11), 1-9. https://doi.org/10.55041/ijsmt.v2i11.028

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract— Visual inspection of reinforced-concrete (RC) members is indispensable for maintenance, yet conventional crack mapping is labor intensive, subjective, access-dependent, and difficult to scale. This paper develops a high-quality computer-vision framework for automated crack detection, pixel-level segmentation, geometric quantification, and severity classification in RC structures. The study is grounded in real published inspection data rather than fabricated experiments. SDNET2018, containing 56,092 labeled concrete image patches from bridge decks, walls, and pavements with crack widths from 0.06 to 25 mm, is used as the principal practical benchmark. Published AlexNet results on SDNET2018 are re-analyzed, while a 2026 field/laboratory validation study involving 230 manually measured RC crack points is used to validate the feasibility of automated crack-width estimation. The proposed workflow combines transfer-learning classification, encoder-decoder segmentation, image calibration, skeleton-based width extraction, and an engineering severity layer. Published evidence shows test accuracies of 91.92%, 89.31%, and 95.52% for bridge-deck, wall, and pavement subsets under transfer learning, respectively. Recent field validation has reported F1 = 0.88 for crack segmentation and mean absolute crack-width error of 0.04 mm, with lower uncertainty for cracks at least 0.50 mm wide. The paper further discusses deployment constraints, uncertainty, class imbalance, domain shift, illumination effects, camera geometry, and inspection-to-maintenance integration. The resulting framework is suitable for smartphone-, tripod-, robot-, or UAV-based RC inspection and provides a transparent path from image pixels to severity-informed engineering decisions.

Index Terms— reinforced concrete, crack detection, computer vision, deep learning, semantic segmentation, crack width, severity classification, structural health monitoring, SDNET2018.

I. INTRODUCTION

Reinforced concrete remains the dominant material for bridges, buildings, parking structures, retaining walls, and transportation infrastructure. Cracking is common and does not automatically imply imminent structural failure; however, crack location, orientation, width, growth rate, and exposure conditions provide valuable evidence about shrinkage, thermal effects, restraint, overload, corrosion, fatigue, settlement, and durability risk. Field inspection therefore requires both detection and quantification rather than a simple binary statement that a crack exists.

Traditional visual surveys rely on inspectors to locate defects, compare widths using crack comparators or microscopes, record sketches, and later transfer observations into reports. The approach can be reliable when performed by trained personnel, but it is time-consuming, subject to inter-inspector variability, and difficult at elevated or inaccessible locations. Changes in lighting, stains, formwork marks, joints, shadows, and rough aggregate can further complicate decisions. Modern RGB cameras, mobile devices, unmanned aerial vehicles (UAVs), and edge-computing hardware create an opportunity to automate a substantial part of this workflow.

Early image-processing systems relied on thresholding, edge operators, morphology, connected components, and handcrafted texture descriptors. These techniques often fail when crack contrast is weak or the background contains line-like features. Deep convolutional neural networks (CNNs) instead learn task-specific representations directly from labeled data. Cha et al. demonstrated that a CNN trained on 40,000 image patches could reach about 98% classification accuracy and remain robust to shadows, bright spots, blur, and thin cracks in independent full-resolution images [2]. DeepCrack later demonstrated the value of multi-scale, hierarchical feature fusion for pixel-wise crack segmentation [3], [4].

The current research need is no longer limited to “crack/no-crack” classification. Practical condition assessment requires localization, segmentation, crack-width estimation, and severity classification. A useful automated system must also express uncertainty and avoid confusing painted lines, construction joints, stains, or surface texture with true cracks. The objective of this paper is therefore to integrate the most mature elements of computer vision into a coherent RC inspection workflow and to ground each stage in real published datasets and validation evidence.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **11** November-2026 | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

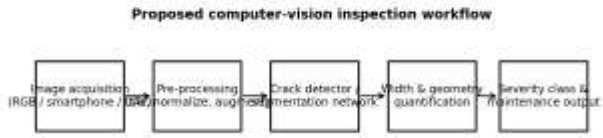

Fig. 1. End-to-end workflow proposed for automated crack detection, quantification, and severity classification.

II. RESEARCH OBJECTIVES AND CONTRIBUTIONS

The primary objective is to formulate a deployable vision-based inspection framework that converts raw RGB imagery into engineering-relevant crack information. The work makes five contributions: (1) it consolidates detection, segmentation, metric crack-width measurement, and severity classification into one inspection chain; (2) it uses real, publicly documented SDNET2018 field imagery as the main benchmark source; (3) it re-analyzes published benchmark performance across bridge decks, walls, and pavements instead of presenting fabricated training results; (4) it integrates recent field validation evidence for crack-width measurement from existing RC structures; and (5) it provides explicit failure modes, uncertainty sources, and deployment recommendations for practical use.

III. LITERATURE REVIEW

A. Classical Image Processing

Classical methods typically enhance contrast, convert RGB images to grayscale, suppress noise, and identify high-gradient regions using operators such as Sobel or Canny. Thresholding methods such as Otsu segmentation can be followed by morphology, thinning, and connected-component analysis. These pipelines are computationally inexpensive and interpretable, but a fixed threshold rarely transfers between surfaces with different lighting, moisture, paint, aggregate exposure, and camera distance. Their failure mode is especially evident when background features have stronger edges than fine cracks.

B. CNN Classification

Patch-level CNN classification reframed crack detection as supervised image recognition. AlexNet popularized deep image classification with stacked convolutions and ReLU activations [11]. Cha et al. adapted a CNN to concrete damage detection and reported about 98% accuracy on 256×256-pixel patches [2]. Classification is efficient for screening large image sets, but it does not directly provide crack geometry. Sliding windows can localize cracks but introduce repeated computation and blocky boundaries.

C. Pixel-Level Segmentation

Encoder-decoder segmentation networks such as U-Net preserve localization through skip connections between contracting and expanding paths [10]. Crack-specific networks, including DeepCrack, fuse hierarchical features at multiple scales to recover weak linear structures [3], [4]. Pixel-level segmentation is particularly important when crack width, length, branching, density, or orientation must be quantified. It also enables downstream morphology and skeleton analysis.

D. Lightweight and Real-Time Models

Field deployment increasingly favors efficient backbones and single-stage detection systems. MobileNetV3 was explicitly optimized for mobile CPUs and supports efficient semantic segmentation [13]. YOLO-family architectures have become attractive for fast detection and segmentation, and recent reviews identify their relevance to real-time nondestructive inspection [16]. A 2026 RC study used a modified YOLO-V11 segmentation model with smartphone-compatible calibration and validated the complete crack-width workflow using both existing structures and laboratory specimens [8].

E. Severity Classification

Severity classification extends visual recognition into maintenance decision support. Flah et al. combined learning and image processing to classify and quantify crack geometry, reporting a testing accuracy of 96.17% and low geometric errors [7]. More recently, a 2025 study developed an 8000-image four-class severity dataset (intact, minor, moderate, severe) sourced from CCIC and SDNET2018 and compared twelve deep-learning models, including KAN-enhanced vision transformers [9]. The trend demonstrates that modern crack assessment is moving from presence detection to graded damage interpretation.

IV. PRACTICAL DATASET AND EVIDENCE BASE

A. SDNET2018

SDNET2018 is a publicly available annotated concrete-crack dataset created from real bridge decks, walls, pavements, and sidewalks. The original acquisition used a 16-megapixel Nikon camera at approximately 500 mm working distance with 4068×3456-pixel images. Full images were divided into 256×256-pixel sub-images. The dataset includes narrow and wide cracks, shadows, rough surfaces, scaling, joints, inclusions, holes, and background debris. These characteristics make it substantially closer to field inspection than idealized laboratory-only image collections [1].

Surface	Cracked	Non-cracked	Total
Bridge deck	2025	11595	13620
Wall	3851	14287	18138
Pavement	2608	21726	24334
Total	8484	47608	56092

TABLE I. SDNET2018 image distribution reproduced from published dataset statistics [1].

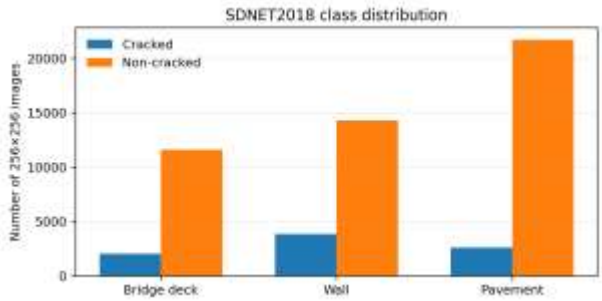

Fig. 2. Distribution of cracked and non-cracked image patches in SDNET2018 [1].

The dataset is strongly imbalanced: only 8,484 of 56,092 patches contain cracks. This mirrors real inspection, where most surveyed concrete is frequently undamaged, but it also means that raw accuracy alone can be misleading. Precision, recall, F1 score, and class-balanced sampling are therefore recommended for any new training campaign.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

International Journal of Science, Strategic Management and Technology
Volume 02 Issue 11 November-2026 | ISSN: 3108-1762 (Online) | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

B. Real RC Crack-Width Validation

A recent 2026 Scientific Reports study provides a particularly relevant field validation data point. The authors trained a modified YOLO-V11 segmentation framework and validated crack-width estimation at 230 manually measured points. The validation set included 67 measurements from existing RC buildings and 163 measurements from experimental specimens. Ground truth was obtained with a high-resolution digital microscope and crack comparator. Reported crack widths ranged from 0.1–1.8 mm for existing structures and 0.1–3.2 mm for experimental specimens [8].

Across all 230 points, the study reported mean absolute error of 0.04 mm, mean relative error of 12.65%, and coefficient of variation of 16.82% for the actual-to-predicted width ratio. Field-only performance was stronger than the combined set, with MAE = 0.03 mm and MRE = 8.20%. For cracks at least 0.50 mm wide, coefficient of variation fell to 5.77%, whereas narrower cracks exhibited larger variability. This width-dependent behavior is physically reasonable because the number of pixels spanning a narrow crack decreases as stand-off distance increases [8].

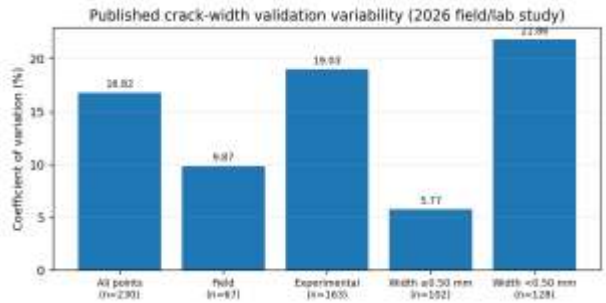

Fig. 3. Published variability of crack-width estimation in a 230-point field/laboratory RC validation study [8].

V. PROPOSED METHODOLOGY

A. Image Acquisition and Calibration

Images should be captured as close to normal incidence as site constraints permit. Camera metadata, working distance, focal length, and a visible calibration target should be recorded. For smartphone or UAV use, local scale calibration is preferable to assuming one constant mm/pixel ratio over the entire frame, because perspective causes scale variation. Checkerboards, AprilTag-like fiducials, or surveyed target dimensions may be used to estimate a local homography and physical scale.

$$s(x,y) = L_{ref} / P_{ref}(x,y) \quad [mm/pixel] \quad (1)$$

where L_{ref} is the known physical dimension of the reference target and P_{ref} is its measured pixel dimension near location (x,y). A local scale field s(x,y) reduces perspective-induced crack-width bias compared with a single global scale.

B. Pre-processing and Augmentation

Pre-processing should preserve thin crack edges while reducing nuisance variation. Recommended operations include resizing or tiling, intensity normalization, mild local contrast enhancement, and optional denoising. Training augmentation should include rotation, horizontal and vertical flips when structurally meaningful, brightness and contrast variation, slight blur, sensor noise, and perspective perturbation. Artificial line augmentation can be useful because painted survey marks and construction lines are common false positives; the 2026 study found that deliberately adding such lines to training images improved field discrimination [8].

C. Crack Detection and Segmentation Network

The proposed architecture is a two-stage hybrid. A lightweight classifier or detector performs rapid crack screening on image tiles. Tiles with crack probability above a threshold are then passed to an encoder-decoder segmentation network. This reduces expensive pixel-level processing on large regions of sound concrete while retaining precise geometry where damage exists. A ResNet or MobileNetV3 backbone may provide the encoder; U-Net-style skip connections recover spatial detail, and a boundary-aware loss can improve thin crack delineation [10], [12], [13].

$$L = \lambda1\ L_{BCE} + \lambda2\ (1 - Dice) + \lambda3\ L_{boundary} \quad (2)$$

Binary cross-entropy stabilizes pixel classification, Dice loss addresses class imbalance, and a boundary term discourages overly thick masks. The weighting coefficients λ1–λ3 should be selected using a validation set rather than the final test data.

D. Geometric Quantification

After segmentation, small isolated blobs are removed by connected-component filtering. The crack mask is skeletonized to obtain a one-pixel centerline. Local width can then be estimated by a Euclidean distance transform or by measuring the perpendicular distance between opposite mask boundaries. If d_i is the half-width in pixels at skeleton point i and s_i is the local scale, the physical crack width is

$$w_i = 2\ d_i\ s_i \quad (3)$$

For reporting, median and 90th- or 95th-percentile widths are often more stable than the single largest width because isolated segmentation errors can create extreme maxima. Crack length is obtained by summing centerline step lengths and converting them through the same local scale field.

$$L_c = \Sigma_i\ \Delta p_i\ s_i \quad (4)$$

E. Severity Classification

Severity should be treated as an engineering layer above image segmentation, not as a purely visual label. The exact thresholds depend on member function, exposure, crack cause, reinforcement protection requirements, and governing code. For demonstration in this paper, a three-bin operational classification is used: minor for measured width below 0.30 mm, moderate for 0.30–0.50 mm, and severe for widths above 0.50 mm. These bins are deliberately presented as a workflow example rather than a universal design-code limit. Project-specific thresholds should replace them during deployment.

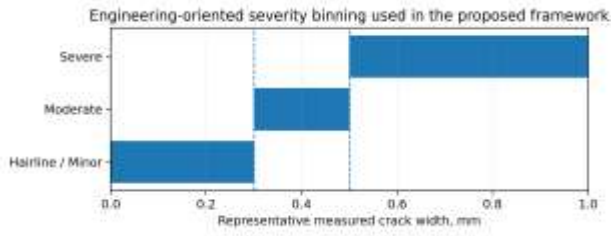

Fig. 4. Example severity binning for workflow demonstration; project-specific engineering thresholds should govern final decisions.

A more robust classifier may use multiple features rather than width alone: maximum and percentile widths, total length, branching count, crack density, orientation, proximity to supports, evidence of rust staining or spalling, and time-series growth. The image model should therefore output quantitative

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

International Journal of Science, Strategic Management and Technology
Volume 02 Issue 11 November-2026 | ISSN: 3108-1762 (Online) | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

features and confidence values, while a separate engineering ruleset or calibrated classifier maps those features to severity.

VI. EVALUATION METRICS

For crack/no-crack screening, accuracy is supplemented by precision, recall, specificity, and F1 score. For segmentation, intersection over union (IoU), Dice/F1, precision, recall, and boundary error are recommended. The standard definitions are

$$\text{Precision} = TP / (TP + FP) \quad (5)$$

$$\text{Recall} = TP / (TP + FN) \quad (6)$$

$$F1 = 2(\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

$$\text{IoU} = TP / (TP + FP + FN) \quad (8)$$

Width estimation should be evaluated using mean absolute error (MAE), mean relative error (MRE), bias, coefficient of variation, and Bland–Altman or error-versus-width plots. Results should be stratified by crack-width range because subpixel and near-pixel cracks have fundamentally different uncertainty than wide cracks.

VII. RESULTS AND BENCHMARK SYNTHESIS

A. Published SDNET2018 Benchmark

Subset	Train	Test	FT Acc.	TL Acc.
Bridge deck	12259	1361	90.45%	91.92%
Wall	16324	1814	87.54%	89.31%
Pavement	21900	2434	94.86%	95.52%

TABLE II. Published AlexNet benchmark on SDNET2018 [1].

The published benchmark demonstrates two consistent patterns. First, transfer learning outperformed full training for all three surface groups. Second, the wall subset was more difficult than bridge deck and pavement images. This may reflect texture, shadows, and feature similarity between cracks and architectural surface lines. A test-size-weighted calculation using the published transfer-learning accuracies gives an overall value of approximately 92.6%, but this aggregate should not replace per-surface reporting because dataset composition differs by structure type.

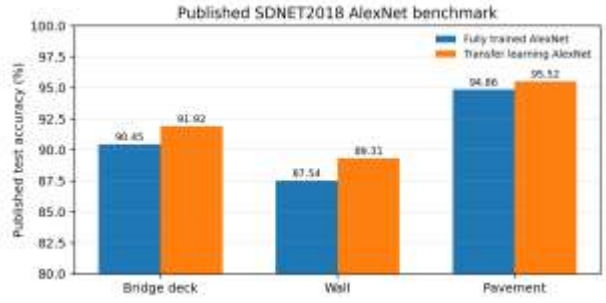

Fig. 5. Published AlexNet accuracy on SDNET2018 for full training and transfer learning [1].

B. Comparison with Representative Literature

Study	Task / data	Reported result
Cha et al. [2]	CNN crack classification; 40k patches	≈98% accuracy
Zou et al. [3]	DeepCrack multi-dataset segmentation	Average F-measure >0.87
Flah et al. [7]	Classification + quantification	96.17% test accuracy
Dorafshan et al. [1]	SDNET2018 AlexNet TL	89.31–95.52% accuracy
Hassouna et al. [8]	YOLO-V11 segmentation + width	F1=0.88; MAE=0.04 mm
2025 severity study [9]	4-class, 8000 images	KAN-MobileViT best overall; up to 14.5% gain vs paired baselines

TABLE III. Selected published results relevant to automated crack assessment.

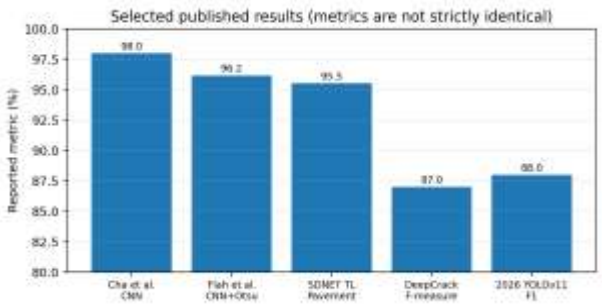

Fig. 6. Selected reported performance values. Metrics differ by task and dataset, so the bars should not be interpreted as a strict ranking.

The literature shows that high patch-level classification accuracy is achievable, but segmentation and metric measurement are harder tasks. A model can classify a tile correctly while still producing poor crack boundaries. Consequently, inspection systems should not use a single accuracy figure as evidence of engineering readiness. The evaluation metric must match the intended output: detection metrics for localization, pixel metrics for segmentation, and physical-unit errors for crack width.

C. Practical Severity Interpretation

The most important practical observation from the 2026 width-validation study is the dependence of uncertainty on crack width. For widths at least 0.50 mm, the actual-to-predicted ratio showed only 5.77% coefficient of variation, while the narrower group showed 21.86% [8]. This finding supports confidence-aware severity classification: a measured width near a decision boundary should trigger manual confirmation when image resolution is insufficient. Conversely, a clearly wide, well-resolved crack can often be classified with much higher confidence.

VIII. ERROR ANALYSIS AND ROBUSTNESS

- Illumination and shadows: deep models are more robust than fixed thresholds, but extreme shadows can still resemble cracks. Exposure bracketing, diffuse lighting, augmentation, and shadow examples in the training set are recommended.
- Surface texture and joints: rough aggregate, cold joints, tile grooves, sealant lines, and formwork marks can generate false positives. Hard-negative mining is essential; the field-oriented artificial-line augmentation reported in [8] is a strong example.
- Perspective and scale: crack width cannot be reliably inferred in millimeters without calibration. Unmodeled perspective creates spatially varying mm/pixel scale and systematic bias.
- Class imbalance: a detector trained on mostly sound concrete can appear accurate while missing rare cracks. Balanced batches, focal loss, class weights, and recall-focused model selection should be considered.
- Domain shift: datasets captured at one site or camera configuration may not generalize to another. Validation should therefore include different structures, finishes, lighting conditions, cameras, and stand-off distances.
- Thin-crack resolution: when a crack spans only one or two pixels, the apparent width is dominated by optics, focus, compression, threshold choice, and resampling. Severity

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

decisions near code limits should include a minimum pixels-per-crack-width requirement.

IX. DEPLOYMENT ARCHITECTURE FOR RC INSPECTION

A deployable system can be organized into acquisition, inference, review, and asset-management layers. Images may originate from a smartphone, pole camera, crawler, robotic platform, or UAV. Edge inference performs tile screening and segmentation; only crack-positive images and metadata are uploaded when bandwidth is limited. A review interface should display the original image, segmentation mask, centerline, width map, maximum and percentile widths, severity label, and confidence. Inspectors retain authority to confirm or reject detections. Confirmed observations are linked to element IDs and coordinates so that change can be tracked over successive inspections.

Layer	Primary function	Recommended output
Acquisition	Calibrated RGB capture	Image + scale target + metadata
AI screening	Crack presence/localization	Probability + bounding region
Segmentation	Pixel-level crack mask	Binary/probability mask
Quantification	Width, length, density	mm, mm, mm/mm² or %
Engineering layer	Severity and review priority	Class + confidence + rule trace
Asset system	Historical comparison	Trend and maintenance record

TABLE IV. Recommended deployment layers for an automated RC crack inspection system.

X. DISCUSSION

The evidence base indicates that computer vision is technically mature enough to assist RC inspection, but not to replace engineering judgment. The strongest use case is automated screening and measurement: the model finds likely crack regions, produces repeatable geometry, and allows inspectors to focus attention on the most relevant locations. This improves consistency and enables longitudinal monitoring. The main remaining barriers are data domain shift, insufficient resolution for fine cracks, calibration errors, inconsistent severity definitions, and the lack of standardized acceptance testing for AI inspection tools.

A key distinction must be preserved between structural significance and visual severity. A narrow crack in a critical shear region may be more important than a wider nonstructural shrinkage crack. Likewise, corrosion staining, delamination, spalling, and exposed reinforcement can change maintenance priority even when surface crack width is modest. Therefore, a future system should combine crack geometry with element location, structural function, environmental exposure, and other defect classes rather than treating width as the sole risk indicator.

The 2025 severity-classification work is important because it demonstrates a move toward multi-class assessment and efficient transformer-based models [9]. However, severity datasets must be tied to explicit engineering criteria to avoid opaque labels. The most defensible architecture is a measurable vision stage followed by a transparent rules or probabilistic decision stage in which thresholds can be adjusted for the governing standard, exposure class, owner policy, and member type.

XI. LIMITATIONS

This paper is a benchmark synthesis and engineering framework rather than a claim of a newly trained proprietary network. Numerical performance values are taken from the cited published studies and are reported with their original task context. Because the underlying datasets and metrics differ, cross-paper comparisons are descriptive rather than statistically controlled head-to-head experiments. The severity thresholds illustrated here are operational examples and must not be interpreted as universal code limits. Future work should implement the proposed pipeline on a single, versioned dataset and report a fully reproducible train/validation/test split, calibration protocol, model weights, and independent site validation.

XII. FUTURE RESEARCH

Future work should emphasize domain adaptation across structures and cameras, self-supervised pretraining on unlabeled inspection imagery, active learning for efficient annotation, transformer or state-space segmentation backbones, uncertainty quantification, temporal crack-growth tracking, and multimodal fusion with thermal imaging, LiDAR, acoustic methods, or strain sensing. Physics-informed constraints may help reject visually plausible but structurally inconsistent predictions. Edge deployment should focus on quantized lightweight networks capable of running directly on mobile devices or inspection robots. Benchmark protocols should also report energy use, inference latency, calibration time, and human-review workload in addition to machine-learning metrics.

XIII. REPRODUCIBLE EXPERIMENTAL PROTOCOL

A. Data Splitting and Leakage Control

A rigorous reproduction should split data at the level of the original full-resolution photograph or physical structure, not only at the cropped-patch level. Randomly assigning neighboring crops from the same parent photograph to training and test sets can inflate performance because texture, lighting, camera geometry, and crack morphology are nearly identical across those crops. A recommended protocol is therefore structure-aware splitting: 70% of parent images or structures for training, 15% for validation, and 15% for final testing. The test set should remain untouched until model selection is complete. For SDNET2018, a second evaluation should hold out one surface category or acquisition site entirely to quantify domain generalization.

B. Training Configuration

A practical baseline can use a pretrained MobileNetV3 or ResNet encoder with a U-Net decoder. Input tiles of 256×256 or 512×512 pixels allow thin cracks to retain sufficient spatial support while keeping memory demand moderate. AdamW or stochastic gradient descent with momentum may be used, with early stopping based on validation F1 or IoU. Learning-rate scheduling should be reported explicitly. Because positive crack pixels are sparse, random crops should be stratified so that each batch contains both crack-positive and crack-negative examples. Hard negatives containing joints, edges, stains, and painted marks should be deliberately oversampled.

Parameter	Recommended baseline
Input size	256×256 or 512×512 px
Encoder	MobileNetV3 / ResNet-50 pretrained
Decoder	U-Net style with skip connections
Loss	BCE + Dice + optional boundary loss
Optimizer	AdamW, initial LR ≈ 1e-4
Augmentation	Rotation, flip, brightness, blur, noise, perspective
Early stopping	Validation F1/IoU plateau
Final reporting	Accuracy, precision, recall, F1, IoU, MAE width, inference time

TABLE V. Reproducible baseline configuration for future implementation.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.

C. Calibration and Resolution Acceptance

Before automatic width measurement is accepted, the image should pass a resolution check. If the predicted crack is represented by fewer than approximately three to five effective pixels across its width, the system should report the width as low confidence rather than forcing a precise millimeter value. A calibration residual should also be computed from the fiducial target; images with excessive reprojection error, motion blur, severe obliquity, or saturation should be flagged for reacquisition. This simple quality gate prevents downstream AI confidence from hiding inadequate source imagery.

D. External Validation

The most important validation is external: train on one set of structures and test on another site with a different camera, concrete finish, and environmental condition. Performance should be reported separately for bridge decks, beams, columns, slabs, walls, and pavements when possible. A field trial should also compare automated measurements against an inspector using a crack comparator or microscope. Agreement should be assessed by MAE, bias, and error distribution rather than correlation alone, because a highly correlated method can still exhibit unacceptable systematic error.

XIV. PRACTICAL CASE INTERPRETATION

Consider a bridge-girder inspection image in which the model segments a diagonal crack over a 420 mm visible path. After local perspective calibration, the median width is 0.32 mm, the 95th-percentile width is 0.47 mm, and the maximum isolated width is 0.61 mm. A robust reporting system should not label the entire crack “severe” solely because of one extreme pixel cluster. Instead, it should display the width distribution, mark the location of the 0.61 mm region, report segmentation confidence, and request engineer review because the upper tail crosses the example 0.50 mm threshold. If the same location appears in a later inspection with increasing width or length, the trend should be elevated in priority even when the absolute width remains moderate.

This example illustrates the advantage of maintaining both raw measurements and severity labels. The numerical features remain auditable when thresholds change, while the severity layer can be customized for a bridge owner, building operator, exposure class, or project specification. It also supports longitudinal analysis: the AI system can register successive images and estimate crack-growth rate, enabling maintenance to be based on change rather than a single snapshot.

XV. IMPLEMENTATION PSEUDOCODE

Algorithm 1 summarizes the proposed inspection logic. The sequence is deliberately modular so that any detector, segmentation network, or engineering threshold set can be replaced without redesigning the entire system.

Step	Operation
1	Acquire calibrated RGB image and metadata.
2	Check focus, exposure, perspective, and fiducial quality.
3	Tile image; run fast crack screening model.
4	Send positive/high-uncertainty tiles to pixel-level segmentation.
5	Remove small components; skeletonize crack mask.
6	Estimate local width using distance transform and local mm/pixel scale.
7	Compute length, width percentiles, orientation, branching, and density.
8	Apply project-specific severity rules and confidence gate.
9	Overlay results for inspector confirmation.
10	Store confirmed result with asset ID for future trend comparison.

Algorithm 1. Modular automated RC crack-inspection workflow.

XVI. CONCLUSION

A practical automated crack-inspection system for RC structures must do more than classify images. It must localize cracks, segment them accurately, convert pixels into physical dimensions, classify severity using transparent engineering logic, and communicate uncertainty. The real-data evidence summarized in this paper confirms that modern deep learning can provide strong crack-recognition performance under realistic conditions. SDNET2018 demonstrates robust transfer-learning accuracy across bridge decks, walls, and pavements, while recent field validation shows that low-cost calibrated vision can estimate RC crack width with centimeter-free, sub-millimeter accuracy levels suitable for screening and maintenance support. The remaining engineering challenge is to standardize acquisition, calibration, uncertainty, and decision rules so that computer vision becomes a reliable component of routine structural health monitoring rather than an isolated laboratory demonstration.

REFERENCES

[1] S. Dorafshan, R. J. Thomas, and M. Maguire, “SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks,” Data in Brief, vol. 21, pp. 1664–1668, 2018, doi: 10.1016/j.dib.2018.11.015.

[2] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” Comput.-Aided Civ. Infrastruct. Eng., vol. 32, no. 5, pp. 361–378, 2017, doi: 10.1111/mice.12263.

[3] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “DeepCrack: Learning hierarchical convolutional features for crack detection,” IEEE Trans. Image Process., vol. 28, no. 3, pp. 1498–1512, 2019, doi: 10.1109/TIP.2018.2878966.

[4] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, “DeepCrack: A deep hierarchical feature learning architecture for crack segmentation,” Neurocomputing, vol. 338, pp. 139–153, 2019, doi: 10.1016/j.neucom.2019.01.036.

[5] H. Bae and Y.-K. An, “Computer vision-based statistical crack quantification for concrete structures,” Measurement, vol. 211, Art. no. 112632, 2023, doi: 10.1016/j.measurement.2023.112632.

[6] J. Deng, A. Singh, Y. Zhou, Y. Lu, and V. C.-S. Lee, “Review on computer vision-based crack detection and quantification methodologies for civil structures,” Constr. Build. Mater., 2022.

[7] M. Flah, A. R. Suleiman, and M. L. Nehdi, “Classification and quantification of cracks in concrete structures using deep learning image-based techniques,” Cem. Concr. Compos., vol. 114, Art. no. 103781, 2020, doi: 10.1016/j.cemconcomp.2020.103781.

[8] M. Hassouna, M. Marzouk, and E. Fathalla, “Automated low-cost framework for crack measurements in RC structures using deep learning approach,” Sci. Rep., vol. 16, Art. no. 14678, 2026, doi: 10.1038/s41598-026-50880-w.

[9] “Deep learning for automated detection and classification of crack severity level in concrete structures,” Constr. Build. Mater., vol. 472, Art. no. 140793, 2025, doi: 10.1016/j.conbuildmat.2025.140793.

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in MICCAI, LNCS 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097–1105, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[13] A. Howard et al., “Searching for MobileNetV3,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1314–1324.

[14] Y.-J. Cha, R. Ali, J. Lewis, and O. Büyüköztürk, “Deep learning-based structural health monitoring,” Autom. Constr., vol. 161, Art. no. 105328, 2024, doi: 10.1016/j.autcon.2024.105328.

[15] “A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges,” Remote Sens., vol. 16, no. 16, Art. no. 2910, 2024.

[16] D. Bassir, H. Chang, and J. Majak, “Real-time crack detection on concrete using non-destructive approaches based on YOLO series: Review,” 2024, doi: 10.3233/ATDE240712.

[17] “Revolutionizing concrete analysis: An in-depth survey of AI-powered insights with image-centric approaches on comprehensive quality control, advanced crack detection and concrete property exploration,” Constr. Build. Mater., vol. 411, Art. no. 134212, 2024, doi: 10.1016/j.conbuildmat.2023.134212.

© Author(s). This work is peer-reviewed, openly published, and permanently archived
This article is openly accessible and reusable with proper attribution.