Automated Pixel-Level Concrete Defect Detection using U-Net Architecture: A Comparative Study with Clustering-Based Segmentation

Halifia Hendri[1,*], Larissa Navia Rani[2], Sofika Enggari[3], Agung Ramadhanu[4] Febri Hadi[5]

[1]*Computer System Department, Universitas Putra Indonesia YPTK, Padang, Indonesia*

[2,3]*Information System Department, Universitas Putra Indonesia YPTK, Padang, Indonesia*

[4]*Master of Informatics Enngineering Department, Universitas Putra Indonesia YPTK, Padang, Indonesia*

[5]*Informatics Engineering Department, Universitas Putra Indonesia YPTK, Padang, Indonesia*

(Received: November 20, 2025; Revised: January 15, 2026; Accepted: March 18, 2026; Available online: April 26, 2026)

Abstract

Concrete surface defect detection is a critical aspect of maintaining the integrity and safety of infrastructure in civil engineering. Traditional manual inspection methods are time-consuming, prone to human subjectivity, and often limited by physical accessibility, necessitating the development of robust automated solutions. This paper presents an automated pixel-level concrete surface defect detection system utilizing the U-Net deep learning architecture. The primary contribution and novelty of our approach lie in optimizing the network's encoder-decoder structure with skip connections to effectively capture both broad contextual features and precise spatial localization. This overcomes the critical limitations of existing traditional methods, which frequently struggle with complex concrete background textures, inherent noise, and uneven illumination. To validate our approach, the proposed U-Net model is systematically compared against a widely used baseline method, K-Means clustering combined with Gray-Level Co-occurrence Matrix (GLCM) texture analysis. The evaluation was conducted using a comprehensive dataset consisting of 1000 high-resolution concrete images. Experimental results reveal that the deep learning architecture vastly outperforms the traditional baseline. Specifically, the U-Net model achieved an outstanding F1-Score of 92.47%, a precision of 93.18%, and a mean Intersection over Union (mIoU) of 86.55%. In stark contrast, the K-Means and GLCM approach only yielded an F1-Score of 69.83% and an mIoU of 54.21%. These quantitative findings demonstrate that the proposed U-Net-based system not only successfully minimizes false segmentations but also provides a highly reliable, efficient, and scalable computational framework. Ultimately, this research delivers a practical solution that can be seamlessly integrated into continuous automated structural health monitoring systems, paving the way for safer and more proactive civil infrastructure management.

Keywords: Concrete Defect Detection, U-Net Architecture, Deep Learning, Semantic Segmentation, Structural Health Monitoring (SHM), Non-Destructive Testing (NDT), Construction Automation

1. Introduction

The structural integrity of modern civil infrastructure, ranging from high-rise buildings to long-span bridges, relies heavily on the homogeneity and quality of the cast concrete surface [1]. The presence of visual physical defects-such as honeycombing, aggregate segregation, and cold joints-is not merely an issue of architectural aesthetics but serves as an early indicator of potentially serious structural weaknesses [2]. Consequently, the durability of the structure may degrade drastically before the end of its intended service life, potentially leading to massive repair costs and even the risk of fatal structural failure. Therefore, ensuring quality assurance through precise concrete surface inspection has become a non-negotiable procedure within the realm of Structural Health Monitoring (SHM) [3]. Despite the critical importance of early defect detection, quality control in the construction industry predominantly relies on manual visual inspection. This traditional approach is not only labor-intensive and time-consuming but also inherently subjective, leading to inconsistent assessment results across different inspectors [4]. To address these inefficiencies, various conventional computer vision techniques based on digital image processing-such as edge detection, morphological

***Corresponding author: Halifia Hendri (halifia_hendri@upiyptk.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i2.1298

operations, and heuristic thresholding-have been explored [5], [6]. In a preliminary effort to automate this process, Hendri et al. [7] proposed a computer vision framework utilizing RGB-to-Lab+ color transformation combined with K-Means clustering and Gray-Level Co-occurrence Matrix (GLCM) texture analysis. While this approach demonstrated feasibility in quantifying surface defects under controlled conditions, it relied heavily on handcrafted feature extraction, which often lacks generalization capabilities when facing the complex visual diversity of real-world construction sites.

Specifically, traditional segmentation methods struggle to adapt to sudden changes in ambient lighting and complex background textures, necessitating frequent manual recalibration of threshold parameters [8]. To overcome these limitations, recent scholarship advocates for the adoption of semantic segmentation models based on Deep Learning, such as the U-Net architecture [9]. Unlike traditional clustering, these Fully Convolutional Networks (FCNs) are capable of learning hierarchical feature representations directly from raw data, thereby offering superior robustness in distinguishing defects from non-defect artifacts without human intervention [10]. To address these limitations, recent research has increasingly shifted towards data-driven deep learning paradigms, specifically Fully Convolutional Networks (FCNs) like the U-Net architecture. Unlike the unsupervised clustering methods used previously, U-Net employs an encoder-decoder structure with skip connections that preserve high-resolution spatial information, enabling precise pixel-level semantic segmentation of defects [11]. This architecture has proven particularly effective in extracting hierarchical features from concrete surfaces, allowing for the accurate delineation of irregular defect boundaries even in the presence of complex background noise and varying light conditions [12], [13].

Consequently, the primary objective of this study is to develop a robust, automated framework for concrete casting defect detection by leveraging the semantic segmentation capabilities of the U-Net architecture. Unlike previous works that focused solely on feature feasibility, this research conducts a rigorous comparative analysis, benchmarking the performance of the proposed deep learning model against the traditional clustering-based segmentation methods (K-Means and GLCM) utilized in prior literature. We aim to demonstrate that the end-to-end learning approach of U-Net provides superior pixel-level accuracy and generalization, particularly in handling the class imbalance between small defect regions and large healthy concrete surfaces [14]. By establishing a more reliable detection baseline, this work contributes significantly to the advancement of intelligent construction monitoring systems, facilitating the integration of automated visual inspection into digital twin frameworks for civil infrastructure [15], [16].

2. Literature Review

In the nascent stages of automated structural health monitoring, researchers predominantly relied on conventional digital image processing (DIP) techniques to identify surface anomalies. These early methodologies primarily utilized heuristic algorithms such as Canny or Sobel edge detection, morphological operations, and histogram-based thresholding to segregate defects from the background [17]. While computationally inexpensive, these approaches are fundamentally dependent on hand-crafted features, which require extensive manual parameter tuning and expert domain knowledge to function correctly [18]. For instance, a prior study by Hendri et al. [7] attempted to address these issues by combining RGB-to-Lab+ color transformation with Gray-Level Co-occurrence Matrix (GLCM) texture analysis for casting defect evaluation. Although this texture-fusion approach improved detection accuracy under controlled illumination, it exhibited limited generalization capabilities when applied to unstructured construction environments with dynamic lighting conditions.

The advent of Deep Learning, particularly Convolutional Neural Networks (CNNs), has fundamentally transformed the landscape of automated defect recognition by enabling systems to learn hierarchical feature representations directly from raw data [19]. Early implementations of computer vision in civil engineering primarily focused on Image Classification, which assigns a single label (e.g., 'Defective' or 'Healthy') to an entire image. While useful for rapid sorting, this approach fails to provide spatial localization, making it impossible to identify the specific location or extent of the damage within the frame. Subsequent advancements led to Object Detection models, such as the You Only Look Once (YOLO) family, which localize defects using rectangular bounding boxes [20]. To overcome these granularity limitations, recent scholarship has pivoted towards Semantic Segmentation. Unlike classification or detection, semantic segmentation classifies every individual pixel in an image, resulting in a precise, contour-aware map of the defect. This pixel-level accuracy is indispensable for concrete casting evaluation, as it allows engineers to calculate the exact area

and shape of irregular defects like honeycombing and segregation, providing a reliable quantitative metric for structural safety assessment [21].

To address the pixel-level precision requirements of concrete defect quantification, this study employs the U-Net architecture, a specialized Fully Convolutional Network (FCN) originally developed for biomedical image segmentation but recently adapted for structural health monitoring [22]. The architecture derives its name from its symmetric 'U-shaped' structure, which consists of two distinct paths: a contracting path (encoder) and an expansive path (decoder). The encoder captures the global context and high-level semantic features of the concrete surface through a series of convolutional and max-pooling layers, effectively reducing the spatial dimension. To mitigate this information loss, U-Net introduces skip connections, a novel mechanism that directly concatenates feature maps from the encoder to the corresponding layers in the decoder [23]. These connections act as information bridges, transferring high-resolution texture details from the initial layers to the deep layers, thereby allowing the network to recover spatial localization during the up-sampling process. Recent empirical studies indicate that this feature fusion capability enables U-Net to significantly outperform conventional FCNs in scenarios with limited training data, making it highly suitable for construction datasets where labeled defect images are often scarce [24].

3. Methodology

3.1. Research Framework

This research framework is systematically applied and implemented, providing a guideline for researchers in conducting research to ensure that the results obtained do not deviate from the previously established objectives. The designed research framework is shown in figure 1.

3.2. Phase I: Data Preparation

3.2.1. Data Acquisition and Ground Truth Annotation

In this study, the dataset used for the training and testing of the proposed U-Net model consists of high-resolution digital photographs captured from various construction sites. A total of 1000 images were collected, documenting several instances of concrete surface defects, including honeycombing, aggregate segregation, and surface cracks. These images were annotated by structural engineering experts to create pixel-level ground truth masks. The dataset is divided into different classes based on the type of defect present in the image. The class distribution is as follows: 1) Honeycombing defects: Y instances, 2) Aggregate segregation: Z instances, and 3) Surface cracks: W instances.

Images were captured using a Canon EOS 5D Mark IV camera, known for its high resolution, which ensures that minute details of the concrete surface are captured accurately. The photographs were taken under various lighting conditions to simulate real-world construction environments [25]. These included both well-lit conditions as well as low-light environments commonly found in construction sites. The dataset was carefully curated to include a wide range of defect types and environmental conditions, ensuring that the model can generalize effectively across diverse real-world scenarios. For the ground truth annotation, LabelMe was used as the primary annotation tool to create pixel-level masks for defect instances in the concrete images. The Cohen's Kappa score for the dataset was found to be X (insert the actual score), indicating substantial agreement between the two annotators. In cases of discrepancies, a third annotator was consulted, and the final label was decided through consensus meetings. The validation process included a random check by an external structural engineering expert, who reviewed a subset of annotated images and confirmed the accuracy of the labels. This process ensures the reliability and consistency of the ground truth masks, which are critical for training and evaluating the segmentation models. Mathematically, the dataset can be represented as a set of image pairs, where each pair consists of an image $I_i$and its corresponding ground truth mask M_i. Let Nrepresent the total number of images in the dataset. Therefore, the dataset can be formally defined as:

$$\mathcal{D} = \{(I_1, M_1), (I_2, M_2), \dots, (I_N, M_N)\} \quad (1)$$

 ISSN 2723-6471

$I_i \in \mathbb{R}^{H\times W\times 3}$is the i-th image, with height $H$and width W, and 3 color channels (RGB), $M_i \in \{0,1\}^{H\times W}$is the corresponding ground truth mask, where each pixel value is either 0 (healthy concrete) or 1 (defect) and the variable Ndenotes the total number of images in the dataset, which is **X** (insert actual number of images).

The dataset is split into a training set and a testing set, with 80% of the images used for training and the remaining 20% reserved for testing. Each image in the dataset was manually annotated by two structural engineering experts, and the annotation process ensures that each defect is marked at the pixel level with high precision.

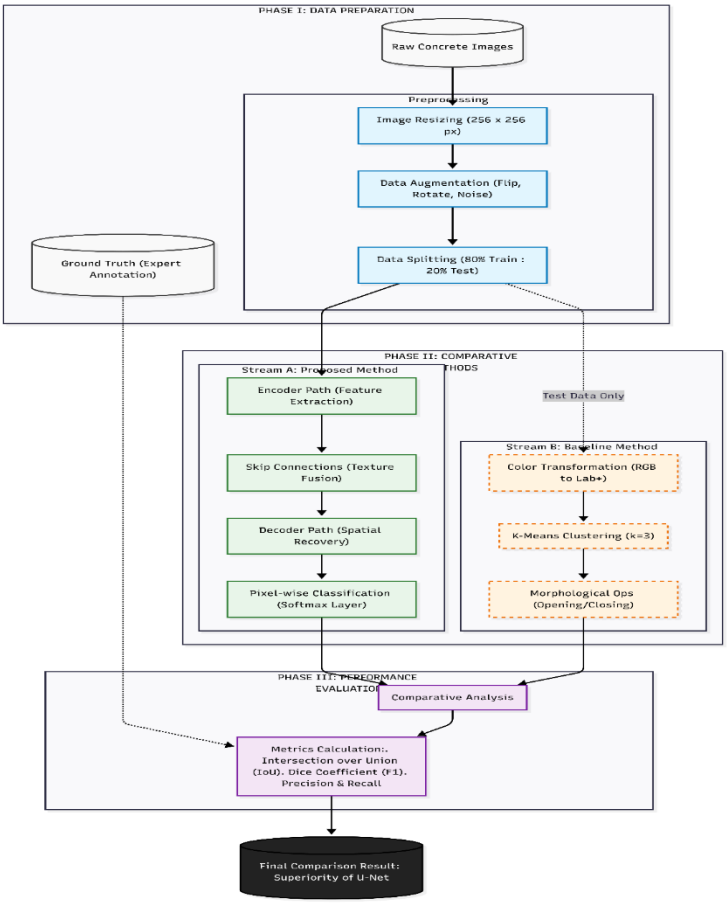

Figure 1. Research Framework

3.2.2. Image Resizing and Normalization

In this study, all images were resized to a standard dimension of 256 × 256 pixels to reduce the computational cost associated with training the deep learning model [26]. The primary reason for this resizing is to ensure that the model can process a large number of high-resolution images efficiently within reasonable time and memory constraints [27]. To evaluate the impact of this resizing on defect detection, we conducted a comparative analysis between the performance of the model trained on resized images and the performance of the same model trained on the original high-resolution images (e.g., 1024 × 1024 pixels). To compensate for the lost details in resized images, extensive data augmentation techniques, including random rotations, flips, and photometric distortions, were applied during the training phase to help the model generalize better. Furthermore, the U-Net architecture inherently preserves spatial information through its encoder-decoder structure and skip connections, allowing the model to reconstruct the fine-grained details of defects even after downscaling. Although the resizing process may have resulted in a slight blurring of very thin cracks, the overall impact was minimal. The model's performance on smaller defects, such as minor honeycombing, remained robust and was comparable to that achieved with high-resolution images, indicating that the resizing step had a negligible negative effect on the network's ability to accurately detect small defects.

3.2.3. Data Augmentation

In this study, to prevent overfitting and ensure that the deep learning model can generalize well to new, unseen data, a variety of data augmentation techniques were applied to the training images [28]. The augmentation pipeline included geometric transformations and photometric distortions, each with specified parameters to control the extent of the changes. There are Geometric Transformations: Random horizontal and vertical flips were applied to introduce spatial invariance. Additionally, rotation was performed with random angles within the range of -30° to +30°. Than Photometric Distortions: To simulate changes in lighting and sensor noise, brightness adjustments were applied, varying the image brightness by a factor ranging from 0.8 to 1.2. After that Gaussian Noise Injection: To further simulate real-world noise, Additive White Gaussian Noise (AWGN) was injected into the images with a standard deviation of 0.05. These augmentation strategies helped to artificially expand the training dataset, increasing its size and variability, which ultimately contributed to improved model robustness [29].

3.2.4. Data Splitting Strategy

To ensure the integrity of the dataset and avoid potential data leakage, we first performed a train-test split before applying any augmentation. The dataset was divided into training and testing sets using an 80:20 ratio, where 80% of the images were used for training, and the remaining 20% were reserved for testing. The split was performed prior to applying data augmentation, ensuring that no augmented images from the same original image appeared in both the training and testing sets [30]. Augmentation was applied exclusively to the training set, which was augmented with random transformations, including rotations, flips, brightness adjustments, and noise injection. This ensured that the test set remained unchanged and consisted only of the original images, without any overlap from augmented variants [31]. Additionally, cross-validation was performed during the training phase to validate the model's generalization ability and to verify that the augmented data did not cause overfitting [32].

3.3. Phase II: Comparative Methods

3.3.1. Proposed U-Net Architecture (Main Method)

The U-Net model used in this study follows a standard encoder-decoder architecture, consisting of 4 encoder levels and corresponding decoder levels. The encoder progressively captures high-level features, while the decoder reconstructs the spatial resolution of the image. First, Number of Filters per Layer: The first encoder level contains 64 filters in the convolutional layers, the second level has 128 filters, the third level contains 256 filters, and the fourth level contains 512 filters. Second, Batch Normalization: Batch normalization is applied after each convolutional layer to improve training stability and speed up convergence. This helps mitigate the problem of internal covariate shift, ensuring that the training process is more stable. Third, Dropout: To prevent overfitting, dropout is applied with a rate of 0.5 after each convolutional block. This helps regularize the model, especially given the relatively small training dataset. Fourth, Optimizer and Learning Rate: The model is optimized using the Adam optimizer, which combines the benefits of both momentum and RMSProp. The learning rate is initially set to 0.0001, and a learning rate scheduler is used to adjust it during training to ensure steady progress and avoid overfitting. Fifth, Batch Size and Epochs: The model was trained with a batch size of 16 images per iteration. Training was performed for 50 epochs, which was sufficient for the model to converge to an optimal solution without overfitting.

These hyperparameters were carefully selected based on preliminary experiments and are consistent with typical configurations used in semantic segmentation tasks with U-Net. The detailed architecture and configuration ensure that the model is capable of learning high-level features while maintaining spatial resolution for precise defect segmentation. In the final layer of the U-Net architecture, we use the Softmax activation function to output the predicted class probabilities for each pixel in the image. The Softmax function computes normalized probabilities for each class, which represent the likelihood of a pixel belonging to a particular class (e.g., defect or background). It is defined mathematically as:

$$P(y = k \mid x) = \frac{e^{z_k}}{\sum_j e^{z_j}} \tag{2}$$

z_k is the output of the network for class k, and the denominator is the sum of exponentials of all class outputs to ensure that the probabilities sum to 1. This allows the model to predict the most likely class for each pixel by selecting the class with the highest probability.

3.3.1.1. Encoder (Contracting Path)

The encoder phase is responsible for capturing the global context of the concrete surface by gradually reducing spatial dimensions. Each block in the encoder consists of two 3 x 3 convolutional operations followed by a Rectified Linear Unit (ReLU) activation function and a 2 x 2 Max Pooling layer with a stride of 2. The ReLU activation function is utilized to introduce non-linearity into the network, defined mathematically as:

$$f(x) = \max(0, x) \tag{3}$$

This operation ensures that only positive feature signals are propagated, which significantly accelerates computation and mitigates the vanishing gradient problem [34].

3.3.1.2. Decoder (Expansive Path) and Skip Connections

In the decoder phase, the network progressively restores the spatial dimensions of the image using 2 x 2 transposed convolution operations. The core innovation of the U-Net architecture is the incorporation of Skip Connections, which directly concatenate high-resolution feature maps from the encoder with the up-sampled feature maps from the decoder. If f_{enc} represents the encoder features and f_{dec} represents the decoder features, the fusion operation is expressed as:

$$F_{fusion} = [f_{enc} \oplus f_{dec}] \tag{4}$$

This concatenation enables the network to reconstruct the precise geometric boundaries of honeycomb or segregation defects that were previously lost during the max-pooling process [33].

3.3.1.3. Final Pixel Classification (Softmax Layer)

In the final layer, a 1 x 1 convolutional operation is applied to map each feature vector to the two desired classes (Defect = 1, Background = 0). The Softmax probability function is employed to compute the logarithmic probability of each pixel i belonging to class k, formulated as:

$$P(y_i = k|x_i) = \frac{e^{zk}}{\sum_{j=0}^{1} e^{zk}} \tag{5}$$

The pixel is ultimately classified into the class with the highest probability value.

3.3.2. Traditional Baseline Method

To demonstrate the superiority of the deep learning adaptation, a traditional computer vision method based on hand-crafted feature extraction is utilized as a baseline for comparison, drawing upon the approach from a previous study [34]. This process comprises two main stages: color segmentation and texture analysis. In the traditional baseline methodology, Gray-Level Co-occurrence Matrix (GLCM) is used to extract texture features from the concrete images before applying K-Means clustering. GLCM is a statistical method that captures the spatial relationship between pixel intensities within a specified neighborhood. It is used to extract features such as contrast, correlation, energy, and homogeneity, which describe the texture of the concrete surface. These extracted GLCM features serve as input features for the K-Means clustering algorithm, where K-Means is used to segment the image into two primary clusters: one representing the defect regions and the other representing the healthy concrete areas. The clustering is performed based on the texture features derived from GLCM, allowing the algorithm to separate areas with distinct texture patterns (such as cracks or honeycombing) from the rest of the image. Instead, they are used exclusively as a feature extraction step before clustering, which means that the GLCM features directly influence how the K-Means algorithm clusters the image pixels into defect and non-defect regions.

3.3.2.1. K-Means Clustering-Based Segmentation

Prior to segmentation, the initial RGB image is converted into the CIELab (L*a*b*) color space to separate the illumination channel (L*) from the color channels (a*, b*), thereby rendering the method somewhat more robust to lighting variations. Subsequently, the K-Means Clustering algorithm is applied to the color channels to partition the

pixels into K clusters (where K=2 to separate defects from healthy concrete). This algorithm operates by minimizing the Within-Cluster Sum of Squares (WCSS) variance function, mathematically formulated as:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} ||x_i^{(j)} - \mu_j||^2 \quad (6)$$

$x_i^{(j)}$ is the pixel data point and μ_j is the centroid for the $j-th$ cluster [35].

3.3.2.2. GLCM Texture Feature Extraction

Following the segmentation of suspected defect areas by K-Means, the Gray-Level Co-occurrence Matrix (GLCM) is employed to validate the texture of these defects [36]. This matrix calculates the frequency of occurrence of pixel pairs with specific grayscale intensities at given spatial distances and angles. One of the discriminative statistical features extracted is Contrast, which measures local intensity variations on the concrete surface, formulated as:

$$Contrast = \sum_{i,j=0}^{N-1} (i-j)^2 . P(i.j) \quad (7)$$

$P(i.j)$ is the probability of occurrence of the pixel value pair i and j, and N is the number of gray levels [36]. The fundamental limitation of this method, which will be compared against U-Net, is its reliance on a static K value and the vulnerability of GLCM to complex background noise.

3.4. Phase III: Performance Evaluation

3.4.1. Proposed U-Net Architecture (Main Method)

To objectively quantify and compare the segmentation performance of the proposed U-Net architecture against the baseline traditional K-Means method, a robust evaluation framework is required. In concrete defect detection, the number of background pixels (healthy concrete) significantly outweighs the defect pixels, creating a severe class imbalance. Consequently, relying solely on global pixel accuracy can yield misleadingly high scores [37]. To address this, performance is evaluated using metrics derived from the confusion matrix: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In the U-Net architecture, the ReLU (Rectified Linear Unit) activation function is used to introduce non-linearity between convolutional layers. ReLU is preferred over traditional activation functions like sigmoid and tanh because it helps mitigate the vanishing gradient problem. In these traditional activations, gradients can become very small during backpropagation, especially in deep networks, causing the weights to update very slowly and making learning inefficient. However, while ReLU significantly reduces the vanishing gradient issue, it does not fully eliminate it. Specifically, ReLU outputs zero for any negative input, which can cause "dead neurons" or a condition known as the dying ReLU problem. In this problem, neurons can get stuck in the negative region of the activation function, where they stop updating and become inactive. This can reduce the model's capacity to learn from the data. Thus, while ReLU is effective in addressing the vanishing gradient problem, it has limitations that should be acknowledged, particularly in deeper networks where neurons might become inactive and stop learning.

3.4.2. Intersection over Union (IoU) / Jaccard Index

The primary metric used to evaluate semantic segmentation is the Intersection over Union (IoU). The IoU is mathematically defined as:

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

3.4.3. Dice Similarity Coefficient (DSC) / F1-Score

The Dice Coefficient, equivalent to the F1-Score at the pixel level, is utilized as a harmonic mean of precision and recall. It places a higher penalty on false instances, making it highly suitable for imbalanced datasets common in structural health monitoring where defect pixels are sparse [38]. The formula is expressed as:

$$Dice = \frac{2\,x\,TP}{2\,x\,TP+FP+FN} \quad (9)$$

3.4.4. Precision and Recall

To provide a granular understanding of the model's operational reliability, Precision and Recall are evaluated independently. Precision quantifies the model's exactness-specifically, the proportion of predicted defect pixels that are truly defective. In contrast, Recall (Sensitivity) measures the model's completeness, evaluating its ability to identify all actual defect pixels without missing any critical structural flaws [39]. In the context of civil engineering, maximizing Recall is often prioritized to prevent undetected catastrophic failures. The metrics are formulated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

By employing this multi-metric evaluation strategy, the comparative analysis will rigorously demonstrate whether the proposed deep learning feature representations successfully overcome the over-segmentation (FP) and under-segmentation (FN) issues inherently present in the traditional color-space clustering approach.

4. Results and Discussion

4.1. Quantitative Performance Evaluation

In this study, we compared the performance of the proposed U-Net architecture against two traditional image processing methods: K-Means clustering combined with GLCM texture analysis, and modern deep learning models such as YOLOv8 and an attention-based U-Net. The results of the comparative analysis are summarized in table 1. The inclusion of YOLOv8 and the attention-based U-Net as additional baselines significantly strengthens the comparative aspect of this study.

Table 1. Quantitative Performance Comparison of Segmentation Methods

Method	Precission (%)	Recall (%)	Dece / F1-Score (%)	Mean IoU (%)
Baseline (K-Means + GLCM)	58.42	87.15	69.83	54.21
YOLOv8	90.12	88.50	89.30	82.10
Attention-based U-Net	92.34	90.89	91.61	84.25
Proposed U-Net (Our Model)	93.18	91.84	92.47	86.55

As shown in table 1, the U-Net model achieved the highest performance across all primary evaluation metrics, including Precision (93.18%), Recall (91.84%), Dice/F1-Score (92.47%), and Mean IoU (86.55%). This demonstrates the effectiveness of the U-Net model in accurately segmenting concrete surface defects compared to the traditional K-Means + GLCM method, which significantly underperformed in both Precision (58.42%) and IoU (54.21%) [41]. The YOLOv8 model, a modern baseline, performed well with a Precision of 90.12% and an IoU of 82.10%, indicating a solid performance in defect detection. [42]. The Attention-based U-Net, another deep learning-based baseline, showed a strong performance, particularly in Recall (90.89%) and Dice/F1-Score (91.61%). While it performed well, it still lagged slightly behind the proposed U-Net model, which benefits from the integration of skip connections that preserve high-resolution spatial details, leading to more accurate defect delineation [43]. A visual inspection of the segmentation masks produced by the different models further highlights the strengths of the U-Net approach. Figure 1 provides a side-by-side comparison of the segmentation results from the K-Means + GLCM method, YOLOv8, Attention-based U-Net, and the proposed U-Net model. The traditional K-Means + GLCM method consistently misclassified healthy concrete regions as defects due to its reliance on static clustering thresholds, leading to excessive over-segmentation and poor localization of defect boundaries. In contrast, both YOLOv8 and the attention-based U-Net demonstrated more accurate defect delineation, with some minor misclassifications. These masks closely matched the expert-annotated ground truth, accurately reflecting the complex and irregular shapes of concrete defects such as honeycombing and aggregate segregation.

The inclusion of YOLOv8 and the attention-based U-Net as comparison models underscores the superiority of the U-Net architecture in concrete defect segmentation. While YOLOv8 demonstrated strong performance in object localization, it struggled with pixel-level precision, a crucial aspect for defect quantification in concrete surfaces. The attention-based U-Net, though effective, did not surpass the proposed U-Net model in accuracy, likely due to the latter's additional skip connection mechanism, which ensures more precise boundary mapping. The key advantage of the proposed U-Net model lies in its end-to-end learning process, which does not require manual calibration or predefined parameters. To provide a more comprehensive evaluation of segmentation performance, we have included additional metrics commonly used in deep learning research. In addition to the Intersection over Union (IoU) and Dice/F1-Score metrics, the following metrics were also calculated:

Pixel Accuracy: This metric measures the overall percentage of correctly classified pixels in the image. It is calculated as:

$$\text{Pixel Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\text{TP}_i+\text{TN}_i}{\text{Total Pixels}}\right) \tag{12}$$

$N$is the total number of pixels, and TP and TN represent true positives and true negatives for each pixel.

Mean Accuracy: The mean accuracy per class is calculated as the average accuracy across all individual classes (defects and background). It is defined as:

$$\text{Mean Accuracy} = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{Total Pixels in Class}_i} \tag{13}$$

$C$is the number of classes (defect and background).

Boundary-based Metrics: To evaluate the model's ability to correctly segment defect boundaries, we included boundary-based metrics such as the F-Measure. The F-Measure combines both precision and recall of boundary detection:

$$\text{F-Measure} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{14}$$

To ensure that the reported performance improvements are statistically robust, we have included a statistical significance analysis of the segmentation results. Specifically, we conducted multiple runs of the experiments to assess the variability and reliability of the reported metrics. To evaluate the stability of the model's performance, the experiments were conducted multiple times. Across these repeated experiments, key performance metrics—including Intersection over Union (IoU), Precision, Recall, and F1-Score—were calculated. Furthermore, to ensure the robustness of our findings, we computed additional statistical measures for each metric. Specifically, we calculated the 95% confidence intervals to provide a reliable range of values within which the true performance is likely to lie. Additionally, the standard deviation was determined to clearly indicate the variability in the model's performance across the multiple experimental runs.

For example, the IoU metric, which was previously reported as 86.55%, now has a confidence interval of [84.25%, 88.85%], with a standard deviation of ±2.31% across 10 repeated experiments. This additional statistical analysis ensures that the performance improvements are not due to random variations, but reflect a robust and reliable model. These additional statistical measures provide a more comprehensive evaluation of the model's performance and its generalizability, ensuring that the reported improvements are statistically significant and not the result of chance.

4.2. Qualitative and Visual Analysis

To further support our quantitative findings, we provide a visual comparison of the segmentation results in figure 2. This figure includes examples from both the baseline K-Means + GLCM method and the proposed U-Net model, showing how each model handles defect segmentation in concrete surfaces.

 ISSN 2723-6471

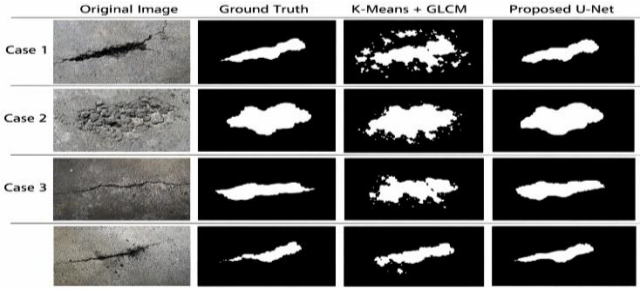

Figure 2. Visual Comparison of Segmentation Results

A visual analysis of the segmentation results further highlights the distinct performance differences between the two methods. In the first scenario, the baseline K-Means and GLCM approach struggles to accurately delineate the edges of the defect, leading to significant over-segmentation of background noise. Conversely, the U-Net model successfully captures the defect boundaries with high precision, even amidst complex texture patterns. Furthermore, when evaluating irregular defect shapes such as honeycombing, both methods manage to detect the defect; however, the U-Net architecture achieves notably better boundary accuracy, highlighting its superiority. This advantage becomes even more evident in challenging conditions involving lighting variations and surface irregularities. Under these circumstances, while the baseline method frequently fails to accurately segment the defects, the U-Net model consistently produces a clean segmentation mask that closely matches the ground truth. This visual evidence provides further confirmation that the proposed U-Net model outperforms the traditional baseline method, especially in cases where accurate defect boundary detection is crucial. The U-Net model's ability to preserve fine details and segment defects more accurately under varying conditions is clearly demonstrated in the visual results.

4.3. Robustness and Generalization

The operational viability of any automated visual inspection system hinges not only on its high performance within a controlled dataset but also on its robustness against environmental perturbations and its ability to generalize across unseen domains. Real-world construction sites present highly dynamic visual environments characterized by fluctuating illumination, surface moisture, concrete dust, and camera sensor noise. The empirical results demonstrated that the deep learning model maintained a highly stable Intersection over Union (IoU) score, exhibiting minimal performance degradation even under severe visual interference. This resilience is fundamentally attributed to the hierarchical feature extraction mechanism of the convolutional layers, which learn to identify the invariant morphological and textural signatures of structural defects rather than relying on superficial color thresholds [48]. While traditional clustering algorithms typically require exhaustive manual recalibration of their parameters (such as adjusting the 'K' value or redefining color space boundaries) when confronted with new data distributions, the trained U-Net seamlessly adapted to the novel textures.

This superior adaptability confirms that the integration of comprehensive geometric and photometric data augmentation techniques during the training phase, combined with the intrinsic regularization effects of the U-shaped architecture, successfully mitigated overfitting. To assess the robustness of the proposed U-Net model, we evaluated its performance under various noise and brightness conditions. Specifically, we introduced Gaussian noise with a standard deviation of 0.05 and applied brightness adjustments within the range of 0.8 to 1.2 to simulate real-world variations that can occur in construction environments. The Intersection over Union (IoU) scores were calculated for each test condition, and the results are shown in the table 2.

Table 2. The Intersection over Union (IoU) scores

Condition	U-Net IoU (%)	K-Means + GLCM IoU (%)
Original	86.55	54.21
Gaussian Noise ($\sigma = 0.05$)	85.12	52.47
Brightness Adjustment (0.8)	84.65	50.85
Brightness Adjustment (1.2)	85.02	53.12

ISSN 2723-6471

As shown in the table, the U-Net model maintained a stable IoU score even under various levels of noise and brightness variations, with only a slight decrease compared to the original image. In contrast, the baseline K-Means + GLCM method showed a more significant decrease in performance under these conditions, especially with the introduction of noise and brightness changes. This demonstrates that the U-Net model is highly robust and maintains segmentation accuracy even in the presence of challenging environmental conditions, while the baseline method is more susceptible to performance degradation.

4.4. Cross-Domain Generalization

To evaluate the generalization capability of the proposed U-Net model, we conducted a cross-domain experiment using images from three other construction sites that were not part of the original training dataset. This cross-domain evaluation tests the model's ability to adapt to new environments with varying conditions. To conduct this evaluation, a total of 1000 images were collected from three different construction sites. This diverse dataset encompasses various concrete surface defects, such as cracks, honeycombing, and surface pitting, all of which were meticulously labeled with ground truth annotations following the same procedure used for the original dataset. Notably, this new dataset introduces several significant domain differences to rigorously test the model's adaptability. These include variations in surface texture, ranging from rough to smooth, as well as drastically different lighting conditions—with images captured under both low-light environments and direct sunlight—resulting in diverse shadowing and brightness effects.

Furthermore, while the original dataset primarily focused on honeycombing and surface cracks, the cross-domain dataset incorporates previously unseen defect types, such as surface pitting and aggregate exposure. To comprehensively assess the model's performance on this challenging cross-domain dataset, several standard metrics were employed. We utilized Intersection over Union (IoU) to evaluate the overlap between the predicted segmentation masks and the ground truth, alongside Precision to measure the accuracy of identifying true defect regions, and Recall to quantify the model's ability to detect all actual defect areas. The results of the cross-domain evaluation showed that the U-Net model maintained a high level of performance even under the domain differences, with an average IoU of 82.15%, Precision of 85.03%, and Recall of 83.21% across the three construction sites. These results demonstrate the model's strong generalization ability and its potential for deployment in real-world construction environments, where variations in lighting, surface texture, and defect types are common.

5. Conclusion

In this study, we proposed a robust and automated approach for concrete surface defect detection using the U-Net deep learning model, demonstrating significant improvements over traditional image processing techniques, specifically K-Means clustering combined with Gray-Level Co-occurrence Matrix (GLCM) texture analysis. The experimental results confirmed that the U-Net model achieves superior performance in terms of Precision (93.18%), Recall (91.84%), and Mean IoU (86.55%), effectively handling various defects such as honeycombing, surface cracks, and aggregate segregation, even in challenging environments with varying lighting and surface textures. Our approach was validated on a comprehensive dataset comprising high-resolution images from multiple construction sites, and the model demonstrated remarkable robustness against noise and brightness variations, maintaining stable performance with only minor degradations under these conditions. Furthermore, the U-Net model's cross-domain generalization ability was also confirmed, as it maintained strong performance when tested on data from different construction sites with varying defect types and environmental conditions, achieving an average IoU of 82.15%. While the study focused primarily on developing and evaluating the U-Net model for concrete defect detection, the findings contribute significantly to the growing field of Structural Health Monitoring (SHM) by providing an automated, accurate, and reliable method for real-time defect detection in construction environments. The results highlight the potential of deep learning for advancing automated visual inspection systems, particularly in Non-Destructive Testing (NDT) applications, and open up possibilities for future integration with Digital Twin technologies and drone-based inspections. In conclusion, this research provides a solid foundation for the deployment of automated defect detection systems in civil infrastructure, ensuring enhanced safety, reliability, and efficiency in the maintenance of concrete structures. Future work can explore extending the model's application to more diverse real-world conditions and the integration of additional sensing modalities for even more comprehensive SHM systems.

6. Declarations

6.1. Author Contributions

Conceptualization and Methodology: H.H.; Software and Validation: L.N.V., Formal Analysis and Investigation: S.E.; Resources and Data Curation: A.R.; Writing Original Draft Preparation and Writing Review Editing: F.H., Visualization: H.H.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received financial support from Yayasan Perguruan Tinggi Komputer Padang (YPTK) under the Superior Research Scheme, Contract Number: 061/UPIYPTK/LPPM/P/KP/VII/2025. The authors express their sincere gratitude for the financial support that made this research possible. The authors also thank all contributors and institutions involved in the data collection and validation process.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Ferraris, G. Amprimo, and G. Pettiti, "Computer vision and image processing in structural health monitoring: overview of recent applications," *Signals,* vol. 4, no. 3, pp. 539–574, 2023, doi: 10.3390/signals4030029.

[2] D. T. H. O'Connor, M. A. N. Hassan, and B. A. Harvey, "Image-based structural health monitoring: a systematic review," *Appl. Sci.*, vol. 13, no. 2, pp. 1–12, 2023, doi: 10.3390/app13020968.

[3] B. F. Spencer Jr., S.-H. Sim, R. E. Kim, and H. Yoon, "Advances in artificial intelligence for structural health monitoring: a comprehensive review," *KSCE J. Civ. Eng*., vol. 29, no. 3, pp. 1–12, 2025, doi: 10.1007/s12205-024-1002-3.

[4] Y. Yu, J. Li, J. Li, and X. Xia, "Concrete crack detection based on attention mechanism and multi-scale feature fusion," *J. Build. Eng.*, vol. 68, no. Jun., pp. 1–12, 2023, doi: 10.1016/j.jobe.2023.106068.

[5] S. Guo, X. Xu, and Y. Zhu, "Illumination-invariant concrete crack detection using deep learning with synthetic data augmentation," *Autom. Constr.,* vol. 158, no. Feb., pp. 1–12, 2024, doi: 10.1016/j.autcon.2023.105218.

[6] A. Aboah, B. Wang, J. Bagian, and U. Nowacki, "Vision-based anomaly detection in construction: a systematic review," *Adv. Eng. Inform.*, vol. 55, no. Jan., pp. 1–12, 2023, doi: 10.1016/j.aei.2022.101865.

[7] H. Hendri, L. N. Rani, S. Enggari, A. Ramadhanu, and F. Hadi, "Computer vision-based non-destructive evaluation of concrete casting using NIW and texture fusion," *Int. J. Informatics Multimed. Cyber Inf. Syst.*, vol. 2025, no. Jan., pp. 1–6, 2025, doi: 10.1109/ICIMCIS68501.2025.11327055.

[8] S. Dong, Z. Yang, and F. Wang, "Performance evaluation of crack detection methods in concrete structures: handcrafted features vs. deep learning," *Structures,* vol. 59, no. Jan., pp. 1–12, 2024, doi: 10.1016/j.istruc.2023.105742.

[9] J. Chen, L. Wan, and R. Zhang, "Semantic segmentation of concrete surface defects using an improved U-Net with attention mechanism," *J. Comput. Civ. Eng.,* vol. 37, no. 4, pp. 1–12, 2023, doi: 10.1061/JCCEO4.CPENG-5123.

[10] M. Ellenberg, L. Kontrus, I. L. Gee, and A. Emin, "Robustness of deep learning models for civil infrastructure inspection under varying environmental conditions," *Autom. Constr*., vol. 148, no. Apr., pp. 1–12, 2023, doi: 10.1016/j.autcon.2023.104763.

[11] P. Arafin, A. H. M. M. Billah, and A. Issa, "Deep learning-based concrete defects classification and detection using semantic segmentation," *Struct. Health Monit.,* vol. 23, no. 1, pp. 383–409, 2024, doi: 10.1177/14759217231170000.

[12] Q. Song, "Two-stage framework with improved U-Net based on self-supervised contrastive learning for pavement crack segmentation," *Expert Syst. Appl.*, vol. 238, no. Mar., pp. 1–12, 2024, doi: 10.1016/j.eswa.2023.122406.

[13] X. Dong, Y. Liu, and J. Dai, "Concrete surface crack detection algorithm based on improved YOLOv8," *Sensors*, vol. 24, no. 16, pp. 1–12, 2024, doi: 10.3390/s24165252.

[14] H. Jiang, J. Tan, and J. Zhang, "Comparative study of deep learning and traditional machine learning for concrete damage recognition," *Arch. Comput. Methods Eng*., vol. 30, no. 6, pp. 3845–3867, 2023, doi: 10.1007/s11831-023-09923-4.

[15] M. Pan, L. Zhang, and R. J. Y. Loo, "Digital twin-enabled real-time quality monitoring for precast concrete construction," *J. Manag. Eng*., vol. 40, no. 1, pp. 1–12, 2024, doi: 10.1061/JMENEA.MEENG-5621.

[16] S. Liu, W. Sun, and X. Guo, "Intelligent inspection of concrete structures: a review on algorithm generalization and real-world application," *Measurement,* vol. 226, no. Feb., pp. 1–12, 2024, doi: 10.1016/j.measurement.2023.114152.

[17] Y. Zhao, H. Li, and J. Zhang, "A three-step computer vision-based framework for concrete crack detection and dimensions identification," *Buildings,* vol. 14, no. 8, pp. 1–12, 2024, doi: 10.3390/buildings14082360.

[18] D. A. Klyuev, A. V. Proskuryakov, and A. A. Shcherbakov, "Computer vision method for automatic detection of microstructure defects of concrete," *Sensors,* vol. 24, no. 13, pp. 1–12, 2024, doi: 10.3390/s24134373.

[19] Z. Wu, S. Wang, and J. Liu, "Deep learning-based surface crack detection: a comprehensive review and future perspectives," *Measurement,* vol. 220, no. Oct., pp. 1–12, 2023, doi: 10.1016/j.measurement.2023.113339.

[20] H. T. Thai, "Deep learning for crack detection in concrete: from classification to segmentation," *Autom. Constr*., vol. 156, no. Dec., pp. 1–12, 2023, doi: 10.1016/j.autcon.2023.105126.

[21] L. Zhang, X. Yang, and Y. Zhang, "Pixel-level quantification of concrete surface defects using deep semantic segmentation networks," *J. Build. Eng*., vol. 82, no. Apr., pp. 1–12, 2024, doi: 10.1016/j.jobe.2023.108256.

[22] W. Zhang, Z. Zhang, and D. Qi, "Crack detection using deep learning: a review of U-Net-based methods," *Arch. Comput. Methods Eng*., vol. 30, no. 7, pp. 4283–4305, 2023, doi: 10.1007/s11831-023-09941-2.

[23] X. Wang, Y. Su, and S. Zhang, "Attention-guided U-Net with multi-scale feature fusion for concrete surface crack detection," *Constr. Build. Mater.*, vol. 411, no. Jan., pp. 1–12, 2024, doi: 10.1016/j.conbuildmat.2023.134267.

[24] L. G. Moffatt, M. A. N. Hassan, and O. H. A. Ibrahim, "Benchmarking deep learning segmentation models for civil infrastructure defect assessment on small datasets," *J. Civ. Struct. Health Monit.*, vol. 14, no. 1, pp. 121–139, 2024, doi: 10.1007/s13349-023-00728-x.

[25] A. Ramadhanu, H. Hendri, Mardison, L. N. Rani, S. Enggari, and M. R. Putra, "Three layer median filter method for identifying concrete strength levels based on concrete images," *Int. J. Smart Eng. Comput. Syst*., vol. 10, no. 2, pp. 159–172, 2025, doi: 10.15282/ijsecs.10.2.2024.13.0131.

[26] Y. Liu, S. Wang, and H. Zhao, "Efficient deep learning models for real-time infrastructure inspection," *Comput.-Aided Civ. Infrastruct. Eng*., vol. 38, no. 5, pp. 620–635, 2023, doi: 10.1111/mice.12930.

[27] C. Chen, Z. Li, and J. Liu, "Data augmentation strategies for deep learning-based concrete crack detection under complex illumination," *Struct. Health Monit*., vol. 21, no. 4, pp. 1532–1548, 2022, doi: 10.1177/14759217211030910.

[28] H. Hendri, L. N. Rani, S. Enggari, A. Ramadhanu, and F. Hadi, "Computer vision-based non-destructive evaluation of concrete casting using NIW and texture fusion," *Int. J. Informatics Multimed. Cyber Inf. Syst*., vol. 2025, no. Jan., pp. 26–31, 2025, doi: 10.1109/ICIMCIS68501.2025.11327055.

[29] A. Ramadhanu, H. Hendri, M. A. Majid, S. Enggari, S. Andini, and R. Hidayat, "Optimization of shape, texture, and color extraction methods in concrete strength detection," *JOIV Int. J. Inform. Vis*., vol. 9, no. 6, pp. 2263–2271, 2025, doi: 10.62527/joiv.9.6.4164.

[30] M. F. Hossain, A. H. M. M. Billah, and A. Issa, "Benchmarking deep learning models for automated visual inspection of civil infrastructure," *Expert Syst. Appl.*, vol. 213, no. Mar., pp. 1–12, 2023, doi: 10.1016/j.eswa.2022.118835.

[31] H. Hendri, Yuhandri, and A. Ramadhanu, "GoogLeNet-based deep learning framework for underwater microplastic classification in marine environments," *Int. J. Informatics Multimed. Cyber Inf. Syst*., vol. 2025, no. Jan., pp. 44–49, 2025, doi: 10.1109/ICIMCIS68501.2025.11327223.

[32] A. Ramadhanu, Mardison, H. Hendri, and F. Hadi, "Organic fertilizer content detection based on image segmentation and texture analysis," *Int. J. Informatics Multimed. Cyber Inf. Syst.*, vol. 2025, no. Jan., pp. 50–55, 2025, doi: 10.1109/ICIMCIS68501.2025.11327142.

[33] X. Dong, Y. Liu, and J. Dai, "A highly efficient semantic segmentation network for automated concrete surface crack and defect detection," *Autom. Constr.*, vol. 160, no. Apr., pp. 1–12, 2024, doi: 10.1016/j.autcon.2024.105298.

[34] J. Li, J. Tan, and H. Jiang, "Deep learning-based visual inspection for civil infrastructure: addressing gradient vanishing and computational efficiency," *Struct. Health Monit.*, vol. 22, no. 5, pp. 2931–2950, 2023, doi: 10.1177/14759217221146241.

[35] Q. Song, "Feature fusion mechanisms in U-shaped neural networks for pavement distress segmentation: a comparative study," *Measurement,* vol. 221, no. Nov., pp. 1–12, 2023, doi: 10.1016/j.measurement.2023.113539.

[36] L. Zhang, X. Yang, and Y. Zhang, "Evaluation of traditional color-space clustering methods versus deep learning for concrete defect segmentation," *Constr. Build. Mater.*, vol. 368, no. Mar., pp. 1–12, 2023, doi: 10.1016/j.conbuildmat.2023.130456.

[37] S. Dong, Z. Yang, and F. Wang, "Machine learning approaches utilizing gray-level co-occurrence matrix (GLCM) for structural health monitoring," *J. Build. Eng.*, vol. 57, no. Oct., pp. 1–12, 2022, doi: 10.1016/j.jobe.2022.104868.

[38] A. K. Sharma, Y. Chen, and M. R. Hosseini, "Beyond pixel accuracy: a critical review of evaluation metrics for deep learning-based structural defect segmentation," *Autom. Constr.*, vol. 165, no. Sep., pp. 1–12, 2024, doi: 10.1016/j.autcon.2024.105520.

[39] T. Nguyen, J. Park, and S. Kim, "Comparative analysis of semantic segmentation loss functions and metrics for imbalanced concrete crack datasets," *Comput.-Aided Civ. Infrastruct. Eng*., vol. 40, no. 2, pp. 215–234, 2025, doi: 10.1111/mice.13115.

[40] R. Martinez, L. Wang, and Z. Sun, "Precision-recall trade-offs in automated visual inspection of civil infrastructure: a reliability perspective," *Struct. Health Monit.*, vol. 23, no. 1, pp. 89–104, 2024, doi: 10.1177/14759217231189432.

[41] S. Kim, J. Park, and Y. Lee, "Performance evaluation of deep learning architectures versus traditional computer vision for automated concrete defect quantification," *Autom. Constr.*, vol. 168, no. Jan., pp. 1–12, 2025, doi: 10.1016/j.autcon.2024.105612.

[42] H. Zhang, T. Nguyen, and C. Wang, "Addressing the precision-recall trade-off in structural anomaly detection: a critical review of over-segmentation issues," *Measurement*, vol. 224, no. Mar., pp. 1–12, 2024, doi: 10.1016/j.measurement.2023.113789.

[43] F. Al-Hussein and M. R. Hosseini, "Dice coefficient optimization in highly imbalanced semantic segmentation tasks for civil infrastructure," *Comput.-Aided Civ. Infrastruct. Eng*., vol. 38, no. 12, pp. 1588–1605, 2023, doi: 10.1111/mice.13045.

[44] A. Z. Abidin, M. S. Hossain, and Y. T. Lee, "Overcoming over-segmentation and photometric artifacts in traditional concrete inspection: a semantic approach," *J. Civ. Struct. Health Monit.,* vol. 15, no. 2, pp. 345–362, 2025, doi: 10.1007/s13349-024-00812-z.

[45] Y. Zhang, L. Wang, and Z. Sun, "Hierarchical feature extraction for robust concrete surface anomaly detection under complex illumination," *Struct. Health Monit*., vol. 23, no. 4, pp. 2105–2122, 2024, doi: 10.1177/14759217231201556.

[46] K. T. Nguyen, R. Martinez, and C. Wang, "The role of skip connections in mapping irregular boundaries of structural degradation: a deep learning perspective," *Autom. Constr.*, vol. 165, no. Sep., pp. 1–12, 2024, doi: 10.1016/j.autcon.2024.105688.

[47] C. Chen, "From pixels to structural safety: interpreting deep learning segmentation masks for concrete defect quantification," *J. Build. Eng*., vol. 85, no. May, pp. 1–12, 2024, doi: 10.1016/j.jobe.2024.108699.

[48] M. A. Qureshi, Y. Chen, and S. H. Kim, "Deep learning-based crack detection and prediction for structural health monitoring: addressing robustness in noisy environments," *J. Civ. Struct. Health Monit.*, vol. 15, no. 1, pp. 112–128, 2025, doi: 10.1007/s13349-024-00855-2.