Journal of Computer Science

Research Article

An Optimized YOLOv11-Based Deep Learning Framework With CNN Feature Enhancement for Tile Crack Detection

Vinod Kumar Pal and Pankaj Mudholkar

Faculty of Computer Applications, Marwadi University, Rajkot, Gujarat, India

Article history
Received: 15-03-2026
Revised: 12-05-2026
Accepted: 06-06-2026

Corresponding Author:
Vinod Kumar Pal
Faculty of Computer Applications, Marwadi University, Rajkot, Gujarat, India
Email:
vinod.csit@gmail.com

Abstract: The tile business is a critical part of the national economic development, as it is one of the areas that provides employment, produces goods, and exports them. Regardless of its significance, there are challenges in the industry that are caused by issues with production, which in most cases is caused by low-quality materials used or by mishandling of the products during transportation. Historically, visitors to the site have been able to detect tile cracks using human eyes, which, in addition to being expensive and time-consuming, can also be unreliable. This paper proposes a trustworthy approach to identifying tile cracks using the YOLOv11 model to solve these challenges. It is a combination of enhanced CNN preprocessing and the YOLOv11 model to enhance the effectiveness of crack detection. It takes advantage of the capabilities of the YOLOv11 model to detect cracks on tiles and aims to use a wide range of tile images in different lighting conditions, textures, and types of defects. The approach uses a powerful tile image analysis approach, and the high detection precision with the bounding box method is 88.30%, and the mask method precision is 88.77% with a small number of false positives.

Keywords: Tiles, Crack Detection, Convolutional Neural Networks, YOLO, Defect Detection

Introduction

India is the third largest tile producer and exporter in the world, which has played a significant role in economic development. In order to maintain this position, there is an urgent need to make sure that Indian tiles are of high quality. However, there are production phases that may lead to defects, including pressing, glazing, and firing, which may cause defects such as cracking (Tian et al., 2025). These tiles must also show structural soundness and surface quality when being marketed. Although in India, tile production and processing have been automated, manual inspection is still being applied to identify imperfections on the surface (Wang et al., 2025). In the recent past, there has been great improvement in using algorithms to detect defects in tiles. Nevertheless, greater accuracy of such automated methods is a huge challenge (Dong et al., 2025). Early detection of cracks in the production cycle is also helpful in reducing material wastage, conserving resources, and improving product quality in the manufacturing process, ultimately reducing the cost of operating the business and enhancing customer satisfaction because of the high-quality products. High-quality standards are critical in the tile industry, and this is facilitated by automatic crack detection systems. The use of the latest deep learning and image processing technologies can transform the process of quality assurance and defect detection.

The most common example of crack damage occurs in faults that are either caused by mishandling, temperature variations, or stress during the manufacturing processes (Wang et al., 2025). The visual inspection method in human beings lacks consistency and is prone to errors; therefore, a few defects can either go unnoticed or be misclassified because of the inspectors being distracted and irritable. Moreover, quantitative measurements and reliable data collection are not available in the manual inspection techniques, and thus, they cannot analyze the knowledge of inspection on defect patterns or improvement of production processes. The modern market involves a lot of industries that use high-quality tiles, and therefore a proper quality control system is needed to detect the flaws (Abbas et al., 2023). These problems indicate the necessity of automatic, accurate, and rapid means of detecting the flaws. Deep learning technologies have various benefits, such as high accuracy, faster processing, and a more stable quality assurance than traditional technology (Shen et al., 2025). The early detection of cracks is important to manufacturing industries since it can minimize the monetary costs and advance the quality of the finished product, which will eventually result in better customer satisfaction. An automatic crack detection system is needed, especially to achieve the high-quality standards of the tile industry (Khanam et al., 2024). (Lin et al., 2024) demonstrate an efficient TinyML-based CNN framework real-time acoustic detection of hollow tile defects, however the framework lacks image-based defect localization, multi-class defect classification, attention-based feature enhancement, and state-of-the-art object detection architectures such as YOLOv11. (Stephen et al., 2021) proposed a Convolutional Neural Network (CNN)-based machine learning framework to distinguish cracked tile surfaces from non-cracked surfaces using a deep learning approach. The framework only performed binary classification and not having crack localization or object detection capabilities. (Lu et al., 2022) introduced a lightweight Convolutional Neural Network (CNN) integrated with a Hand-Crafted Feature Enhancement (HFE) module. The proposed work advantages are Hybrid feature extraction strategy, Lightweight network architecture and improved small-defect detection, although Limited feature adaptability and Limited scalability for multiple defect classes have not discussed.

Motivation

The latest one-stage object detection models, YOLOv11, have excelled in maintaining a balance between detection accuracy and computational efficiency. Compared to previous YOLO variants like YOLOv5 and YOLOv8, YOLOv11 provides more reliable feature representation and multiscale detection, which helps in detecting more granular structures with weaker contrast, such as tile cracks. Besides, its modular architecture eliminates the need to alter the existing internal structural organization for integration with outside preprocessing methods. These characteristics make YOLOv11 a suitable baseline for the proposed study, where the objective is to improve the quality of the input features while not compromising the efficiency and robustness of the detection network.

The effective approach explores deep learning strategies for the creation of reliable and automatic crack detection in tiles in order to overcome the present constraints and set new benchmarks for quality control in this sector:

1. To annotate the tile cracks using Robo Flow
2. Evaluate the performance of the effective approach in crack detection
3. Examine the performance of the effective approach in parameter utilization in training and testing
4. Automated Crack Detection

Literature Review

In this section, various approaches have been explored in the area of crack detection. We have critically reviewed the work done by the researchers in the past to identify the gaps and potential directions for enhancement, such as best precision and faster processing.

Deep Learning

Deep learning can detect defects on tiles. Deep-learning-based automated models are capable of identifying the tiniest or most peculiar cracks that can be overlooked in the process of traditional visual examination. With the application of these technologies, human error is minimized, and the inspection process is quicker and more accurate, which contributes to the increase in the reliability of the results. YOLO is an object detector that can offer an effective implementation of automated crack detection. This method employs only a single neural network to generate bounding boxes and class probabilities of entire images in a single pass, which enables it to run in either real-time or almost real-time. After being trained on a large set of annotated pictures of cracks, the model can detect the presence, location, and approximate size of cracks very fast and with good accuracy.

Seminal Contribution in Crack Detection

The study carried out by Karimi et al. (2024) applied the YOLO model in order to automatically identify tiles with the poorest quality. They analyzed a set of 5000 images that had cracks in tiles and had an accuracy of 72 percent with multiple classifications and 97 percent with binary classifications. Lin et al. (2024) also addressed the issue of identifying floor tile sub-surface defects to enhance quality control in residential buildings. They proposed an AI Diagnostic stick (AID-Stick) to detect tile defects that had a validation accuracy of 97% and real-world accuracy of 81.25% in comparison to the traditional methods. Wang et al. (2024) have suggested another method called CTDD-YOLO that will eliminate small flaws in the system of identifying defects in the tiles. This method was tested on a defect detection dataset and compared to YOLOv8n; the method decreased the number of parameters while increasing the mean Average Precision (mAP) by 7.2 percentage points.

Coskun et al. (2022) presented a machine vision-based framework to classify five common surface defects, namely cracks, scratches, pores, spots, and flecks, in a real-time industrial environment. The proposed framework was not investigated on unseen defect types and limited localization capability.

DOI: 10.3844/jcssp.2026.2411.2424

Pan et al. (2024) used a modified version of the YOLOv5 algorithm to detect surface imperfections on ceramic-coated disks and was successful in overcoming the limitations of small defects, size differences, and poor accuracy of fault detection. This enhanced algorithm showed better results for fracture damage, slag, and spot faults, with improved results of 0.2, 4.7, 5.4, and 1.9, respectively. At the same time, Huang et al. (2024) used a U-shaped convolutional neural network named DEU-Net to identify the surface defects of magnetic tiles. The model gave an F1 value of 74.89 and MIoU of 66.94, which was due to the parameters adopted. In the study Zhou and Tiong (2024), in predicting different defects, the authors focused on crack and spalling defects using a convolutional neural network model. According to their study, they reported accuracies of 0.89 and 0.85 for crack and spalling defects, respectively.

Cumbajin et al. (2023) presented a more sophisticated approach to identifying flaws in industrial tile products with the help of deep learning models. The principle was tested and accepted by a leading company with its high-quality stoneware and tableware. The solution had a fairly high F1-Score of 97.29% and an accuracy of 98. Two machine learning approaches were investigated in Carvalho et al. (2023), and the results of automated visual inspection of dinnerware were reported on the basis of the data presented in the VAI project. Classification of dots and cracked defects was also strong, displaying the F1-score of about 86 percent and the accuracy of about 91 percent and 89 percent, respectively. A study by Ibrahim et al. (2023) employed image processing as the method of automated classification of broken tiles targeting three categories of defects (cracks, corners, and edges). The research produced significant accuracy, sensitivity, and specificity, and the values were 93.34, 89.93, and 95, respectively. The article described a model, namely a deep learning-based system for identifying crack defects in tiles. Demonstrate pre-trained Deep Convolutional Neural Network (DCNN) based on AlexNet to detect invisible cracks. The proposed system used indirect crack detection, it cannot determine the exact location, size, orientation, or shape of cracks. (Zhu et al., 2022) proposed a novel Cross Patch Attention Module (CPAM) to improve defect detection. The proposed was evaluated using YOLOv7, YOLOv5, YOLOv4, recent models have not investigated. Gungor (2022) presented a Gradient-Based Surface Defect Detection for automatic inspection of tiles. The work outperformed traditional edge detection methods for ceramic tile surface inspection, the framework remains limited by its reliance on handcrafted image processing, manual parameter tuning, and binary defect detection.

Comparative Analysis of Related Work

A comparative analysis of tile defect detection is provided in Table 1, with promising results of diverse deep learning models and datasets; however, there are still a number of research gaps. Other models like YOLOv7, YOLOv8n, and CNN-based models are highly accurate but have the issue of being unable to precisely identify the size of the defect, particularly width and height.

Table 1: Comparative Analysis of Related Work

Related Work	Year	Algorithm	Accuracy/Precision	Merits	Demerits
Tian et al. (2025)	2025	YOLOv11	0.607 mAP	Increase the accuracy for concrete crack defect detection	Proposed work would be enhanced for complex datasets.
Karimi et al. (2024)	2024	YOLOv7	72%	Able to detect various defects.	No proof of calculation of the width and height of the defect.
Wang et al. (2022a-b)	2024	AI diagnostic Stick (AID-Stick)	81.25%	Works efficiently on hollow defects in floor tiles.	The proposed method has not been optimized in operational environments.
Wang et al. (2024)	2024	YOLOv8n	78.9%	Achieved good accuracy with low parameters and efficient computational time.	Not suitable for various point defects.
Pan et al. (2024)	2024	YOLOv5s	83.4 %	Good detection performance	Has to be tested on validation in a specific application.
Huang et al. (2024)	2024	Convolutional neural network	MIoU -66.94	Classification accuracy is higher.	The proposed model would be further enhanced in terms of accuracy.
Zhou and Tiong (2024)	2024	CNN Model	Recall (0.89)	The model is able to detect crack and spalling defects.	The proposed model would be further enhanced in terms of accuracy.
Carvalho et al. (2023)	2024	YOLOv8n	Mean accuracy and precision improved by 7.2 %	Able to detect tiny (minor) flaws in tiles.	Would be used for various real-world industrial applications.

DOI: 10.3844/jcssp.2026.2411.2424

Certain models are not very good at detecting minor defects, and hence can hardly be applied to real-world industry with edge devices. Although these models have not only enhanced the recall of minor white points, they are less efficient in accurately recalling black points. The inconsistency of the model accuracy implies that additional refinement is required to fit various types of tiles, operating conditions, and types of defects. In addition, it is still hard to improve the model in other environmental conditions and minimize the complexity of computations without reducing precision.

Key Considerations

The development of our effective and streamlined YOLOv11-based model to detect tile cracks was due to a series of key considerations to guarantee its industrial use, high precision, and robustness. All these were significant in our architectural choices and training patterns:

1. We investigated the various versions of YOLOv11 (nano, small, medium, large, XL) and found the m version to have the highest precision for defect detection
2. We performed the proper augmentation to avoid overfitting so the proposed model will be generalized
3. The inference speed has been tested on all versions of the YOLOv11 family
4. False Positive Rate: Minimizing false alarms is critical to avoid discarding good tiles and reducing operational costs

Methods

In this section, the architectural design and the methodological framework of the proposed YOLO-CNN hybrid architecture that detects defects in tiles will be described. It describes the combination of the object detection functions of YOLO with CNN-enhanced feature classification to achieve better detection performance.

Figure 1 shows the overall architecture of the proposed framework. The two principal components constitute the pipeline:

(i) A dedicated CNN-based preprocessing module
(ii) The baseline YOLOv11 detection network

In contrast to the standard YOLOv11 design, which takes raw input images directly through the internal CNN backbone, the proposed solution adds another lightweight CNN block before the detection phase. In particular, the module is made up of a “convolutional layer with 16 filters of size 3×3, followed by a convolutional layer with 32 filters of size 3×3.” The input image $I \in R$ H × W ×3 is first passed through a Conv layer (3×3 kernel, 32 filters, stride 1, and padding 1), Batch Normalization, and a ReLU activation to extract low-level features, such as edges and textures. This is followed by a Conv layer again (3×3 kernel, 64 filters, stride 1, padding 1) with Batch Normalization and ReLU activation, enabling the spatial representation of features at a deeper level.

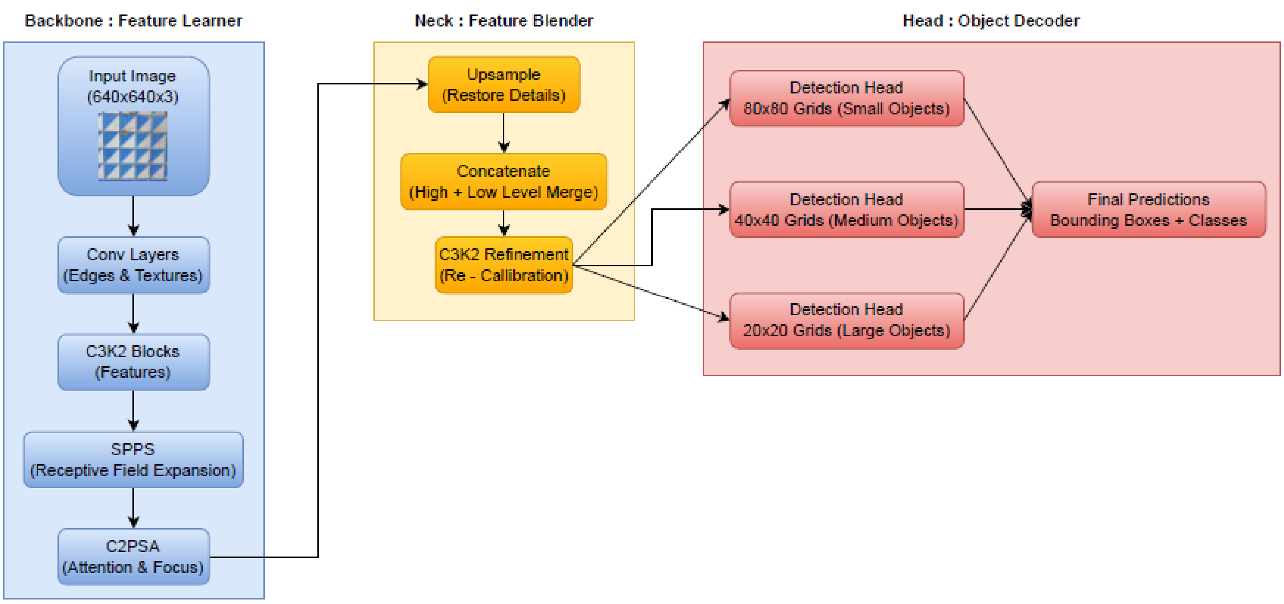

Fig. 1: Proposed Architecture

To further enhance salient defect regions, a third Conv layer (1×1 kernel, 64 filters) is introduced to perform channel-wise feature recalibration and alleviate redundancy. A Max Pooling (2×2, stride 2) is optionally added to reduce spatial dimensions and is more robust against noise. Finally, a Conv layer (3×3 kernel, 3 filters, stride 1, and padding 1) is tasked with reconstructing the enhanced feature map back into a three-channel representation, suitable for YOLOv11 input requirements. The output of this module is a proximized version of image representation feeding into YOLOv11 for multi-scale feature extraction and object detection, which demonstrates that the internal structure of YOLOv11 remains unrevised, with the suggested case constituting an 'external' modification with the intention of upgrading input quality, hence delivering the prospect of optimizing localization and classification of small-sized and low-contrast defect instances without risk of increasing complexity associated with the underlying detection infrastructure.

The "C3K2" block is inferred to mean a Cross-Stage-Partial module with 3 convolutional layers and a 2x2 Kernel. The input tensor X is split into two parts along the channel dimension: $X=[x1, x2]$ where $x1 \in RH\times W\times\alpha Cin$, $X2 \in RH\times W\times(1-\alpha)Cin$, considering $\alpha=0.5$ for an equal split. Features need to be extracted with two convolutions using the:

$$Z1=\sigma(BN(Conv3\times3(1)(X1)))$$

$$Z2=\sigma(BN(Conv3\times3(1)(X1)))$$

$$Z3=\sigma(BN(Conv2\times2(3)(X2)))$$

The output of the previous phase is used for feature aggregation, so we achieve the desired output dimensionality. The overall detection loss function Ltotal for the YOLOv11 model, which incorporates the C3K2 features, is a composite of Equation 4:

$$Ltotal=\lambda boxLCIoU+\lambda objLobj+\lambda clsLcls$$

Where LCIoU: Complete IoU loss for bounding box regression, Lobj.

Objectness loss, Lcls: Classification loss, and λ: Weighting coefficients. As a result, this block is said to be computationally efficient by contributing the minimum loss.

The C2PSA block is an advanced building block that integrates channel-splitting for efficiency with a polarized self-attention mechanism $U=X2\in RH\times W\times C$ to model long-range dependencies crucial for detecting large-scale texture anomalies and repetitive pattern defects on tiles. The C2PSA block incorporates dual Partial Spatial Attention-enhanced branches $Z2=\Psi PSA$ arranged in parallel branches that process separate portions of the feature map before being merged through concatenation. Then, compute the spatial attention map by aggregating information across the channel dimension, preserving the full spatial resolution. Equation 5:

$$A_{s}=Softmax\ (Ks^{T}Qs/\sqrt{C})\in R^{(H.W)\times(H.W)}\ F_{s}=VsAs^{T}\in R^{C\times(H.W)}$$

$$F_{s}=Reshape\ (F_{S})\in R^{HXWXC}$$

The average and max pooling are applied for better detection accuracy using the following formula:

$$A_{avg}=AdaptiveAvgPool2d\ (Fs')\in R^{1\times1\times C}$$

$$A_{max}=AdaptiveMaxPool2d\ (Fs')\in R^{1\times1\times C}$$

The overall objective function for the C2PSA model with:

$$L_{total}=\lambda boxLCIoU+\lambda objLObjFocal+\lambda clsLClsBCE+\lambda dfL_{DF,}$$

Which leads to enhancing the sensitivity.

Methodological Flow

The crack detection system presented in Fig 2 was carried out by loading the Kaggle dataset of tile images at once. Rob flow was used to make annotations; bounding boxes were drawn around the areas of the crack to allow the use of localization. YOLOv11 was trained using five variants of the annotated dataset loaded in the YOLOv11 format. The Yolo files of the weights were imported and optimized in the YOLOv11 format. The trained model was good at detecting objects, which is the ability to recognize cracks in tiles.

Optimized Algorithm Approach

1: TILECRACKDETECTION procedure (Images)
2: // Data Preprocessing
3: for each image I in Images do
4: Resize I to 640 x 640
5: Normalize pixel values to [0, 1]
6: Perform data augmentation: mosaic, mixup, rotate, flip
7: Annotation to YOLOv11 Converter.
8: end for
9: Split Images into train, val, and Test (70:20:10)
10: // Model Initialization
11: Load pre-trained YOLOv11 model
12: Set the hyper parameters: Image size = 640, batch size = 32, epoch = 20, optimizer = Adam, learning rate = 0.01.
13: // Model Training
14: Train M on the train set and validate on an epoch basis.
15: Evaluation metrics: mAP, mAP: 0.95, mAP: 0.5, precision, recall
16: Early stopping should be used when validation metrics are not increasing.
17: // Model Evaluation
18: if mAP@0.5 < threshold then
19: Hyper parameter optimization.
22: Improvedly configured retrain model.
23: end if
24: // Final Model Testing

DOI: 10.3844/jcssp.2026.2411.2424

25: Create a confusion matrix and elaborate a performance report.
26: // Inference Pipeline
27: Test: given Test image, T.
28: D Inference (M, T)
29: Filter D with confidence > 0.5
30: Create bounding boxes and labels along with the label of the classes and confidence.
31: end for
32: Produce detection reports containing: total amount of crack area, total amount of crack length, Confidence levels by detection.
33: // Result Visualization
34: // Remote access Web API deployment.
35: end procedure

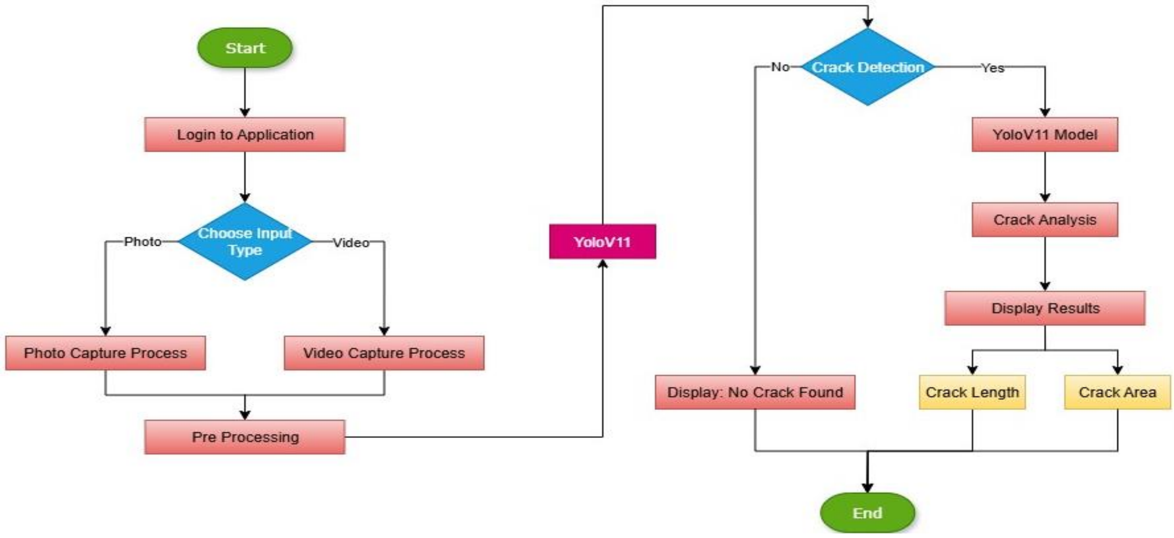

Fig. 2: Methodological Flow

The above hyper parameters are selected after empirical validation and standard practices in YOLO-based training, and were kept consistent across all experiments to ensure fair comparison.

Dataset

The Dataset (Fig. 3) employed in this paper consists of surface pictures of concrete tiles having visible cracks. The pictures depict a broad range of cracks, including hairline cracks, open fractures, and discontinuities in odd configurations, at different lighting and textural settings. This variability indicates that the dataset is reflective of actual world contexts and thus the model is stronger to the variations in appearance of the surface. The dataset was filtered to include various orientations, densities of cracks, and background textures, thus giving a sufficient number of hard cases to train the proposed detection framework and evaluate it. The dataset used in this study was acquired from a publicly available source. The dataset source is (https://www.kaggle.com/datasets/arnavr10880/concrete-crack-images-for-classification), comprising two categories: crack (2000) and non-crack (2000) surface images of tiles. The dataset includes variations in illumination, texture, crack thickness, and orientation to reflect real-world conditions. These were divided into 70% training, 20% validation, and 10% testing.

Fig. 3: Dataset Used

In order to increase the strength of the model and reduce overfitting, some data augmentation methods were used throughout the preprocessing stage, such as random rotation, horizontal and vertical flipping, scaling, brightness and contrast manipulation, Gaussian noise addition, and cropping.

Annotation Process

The dataset images were manually annotated to localize crack defects using bounding boxes. For each image, crack areas were carefully localized inside bounding boxes of arbitrary rectangular shapes following the YOLO annotation format normalized by reference to image dimensions. For thin and elongated crack structures, care was taken to only enclose the cracks and avoid including most of the adjacent background. Non-crack images were labeled without bounding boxes and assigned to the background class. Since 2000 images in each class were used for annotation, there is no imbalance in the classes.

A manual verification step was performed to examine the labeling consistency with regard to low-contrast and ambiguous crack patterns to ensure annotation quality. These Misannotations were either corrected or removed during the process. The resulting annotated Dataset was then used for training, validating, and testing the detection model.

Results and Discussion

Qualitative Analysis

After data pre-processing and annotation, the YOLOv11 model is used for training. Five variants of the YOLOv11 model are used for this purpose: YOLOv11l, YOLOv11m, YOLOv11n, YOLOv11s, and YOLOv11xl, with a learning rate of 0.01 and training epochs set to 20. The image size taken for this study is 640×640.

The evaluation of the performance of a crack detection model is conducted in several perspectives. Fig. 4 demonstrates the estimated distribution of scores of crack and background classes at various classification thresholds (0.3, 0.5, and 0.7) and the variation of the two-class separation. Fig. 5 monitors the main performance indicators -Precision, Recall, F1-score, and Accuracy of the models at varying decision thresholds and therefore exemplifies the trade-offs in the model performance. Fig. 6 is concerned with the concentration of anticipated probabilities of crack and background samples at one threshold, 0.5. Fig. 7 gives the confusion matrix with the number of classes and percentages, and the general measures like Accuracy (69.4%) and Precision and Recall under each of the classes of the word "crack". Lastly, Fig. 8 conceptually illustrates how true labels and predicted labels are related, which supports the logic of classification of the model.

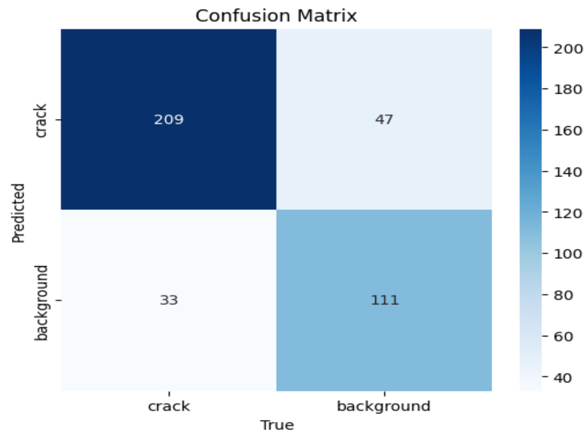

Fig. 4: Confusion Matrix of YOLOv11n

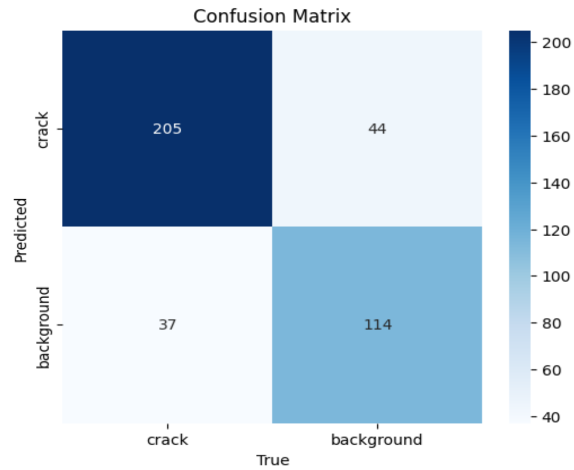

Fig. 5: Confusion Matrix of YOLOv11s

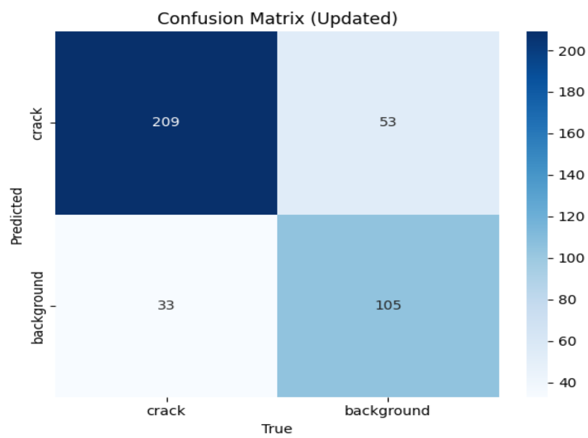

Fig. 6: Confusion Matrix of YOLOv11m

DOI: 10.3844/jcssp.2026.2411.2424

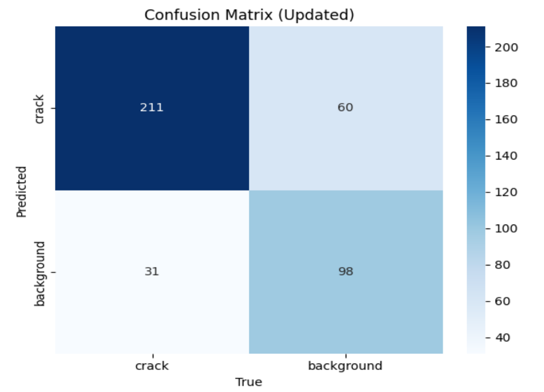

Fig. 7: Confusion Matrix of YOLOv11l

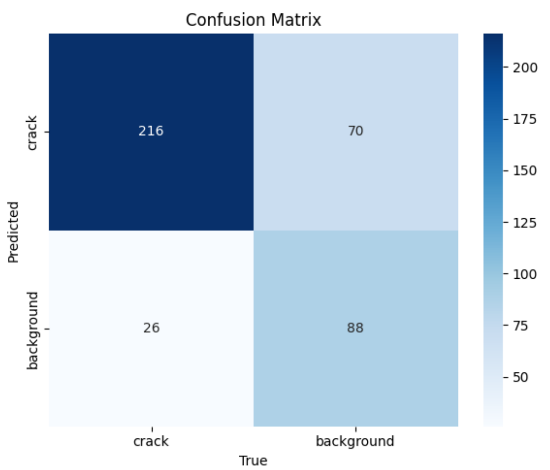

Fig. 8: Confusion Matrix of YOLOv11xl

Quantitative Analysis

As given in the above tables (Table 2) and (Tables 3-4), the performance of the five YOLOv11 variants (YOLOv11l, YOLOv11m, YOLOv11n, YOLOv11s, and YOLOv11xl) in object detection tasks has been evaluated and tested based on various key metrics. YOLOv11m achieved the highest precision (0.88304 and 0.88779) among the models, indicating its effectiveness in correctly identifying objects. YOLOv11xl followed closely with a precision of 0.88005and 0.87089. YOLOv11l, YOLOv11n, and YOLOv11s exhibited slightly lower precisions compared to the other two models. YOLOv11n demonstrated the highest recall (0.80992 and 0.80165), followed by YOLOv11s, YOLOv11m, YOLOv11l, andYOLOv8xl.

YOLOv11m achieved the highest mAP50 (0.87601 and 0.86967) among the models. YOLOv11n, YOLOv11s, YOLOv11l, and YOLOv11xl exhibited slightly lower mAP5 compared to the YOLOv11m models.

YOLOv11xl achieved the highest mAP50-95 (0.70812and 0.47579) among the models. YOLOv11n YOLOv11s, YOLOv11m, and YOLOv11l exhibited slightly lower mAP5-95 compared to the YOLOv11m models.

However, YOLOv11m achieved the box_loss (0.68056 and 0.74271), YOLOv11xl achieved seg_loss (1.60377 and 1.42841), YOLOv11n achieved cls_loss (0.8644 and 0.92671), and YOLOv11m achieved dfl_loss (1.21635 and 1.13797), which are the limitations of the model.

Table 2: Performance comparison of YOLOv11 model variants for ceramic tile crack detection and segmentation using precision, recall, and mAP metrics

MODEL	Time	Metrics /precision(B)	metric /recall(B)	metrics/ precision(M)	metrics /recall(M)	Metrics /mAP50(B)	metrics /mAP50-95(B)	Metrics /mAP50(M)	metrics/ mAP50-95(M)
n	263.985	0.84098	0.80992	0.83462	0.80165	0.85380	0.66956	0.84201	0.47353
s	298.520	0.83296	0.80579	0.82869	0.80165	0.86468	0.67575	0.84018	0.46535
M	505.544	0.88304	0.76860	0.88779	0.77273	0.87601	0.69254	0.86967	0.47579
L	608.632	0.84689	0.77709	0.83800	0.76953	0.86906	0.70107	0.83211	0.46049
XL	970.317	0.88005	0.78828	0.87089	0.78045	0.86887	0.70812	0.85053	0.46156

Table 3: Validation performance comparison of YOLOv11 model variants using box loss, segmentation loss, classification loss, and DFL loss metrics

MODEL	train/ box_loss	train/ seg_loss	train/ cls_loss	train/ dfl_loss	val/ box_loss	val/ seg_loss	val/ cls_loss	val/ dfl_loss
n	0.6375	1.40281	0.8644	1.16574	0.71298	1.3017	0.92671	1.08295
s	0.6136	1.42964	0.58722	1.15209	0.72649	1.31565	0.73735	1.1052
M	0.68056	1.52158	0.6315	1.21635	0.74271	1.31385	0.71548	1.13797
L	0.62429	1.53392	0.61003	1.17875	0.67889	1.37477	0.72416	1.10384
XL	0.67322	1.60377	0.67688	1.22203	0.66916	1.42841	0.73368	1.11156

Table 4: Learning rates for the optimizer parameter groups across the five YOLOv11 model variants

MODEL	lr/pg0	lr/pg1	lr/pg2
n	4.25E-05	4.25E-05	4.25E-05
s	4.25E-05	4.25E-05	4.25E-05
M	4.25E-05	4.25E-05	4.25E-05
L	4.25E-05	4.25E-05	4.25E-05
XL	4.25E-05	4.25E-05	4.25E-05

The figures below, namely Fig. 9 (i), (ii), (iii), (iv) and (v), are the visualizations of the metrics evaluation graphs. YOLOv11n model was tested using 20 epochs. During training, the loss is gradually reduced with increasing epochs, and this shows that the model is performing better. Decreasing trends are observed during validation, and this implies that the model is well generalized on validation data. The reduction of all training losses in the YOLOv11s model points to the improvement of this model in the course of training. There is a spike in DFL loss that is briefly followed by stabilization, indicating model adjustments at the initial training. Validation losses exhibit the same downward trend as the training losses, indicating that they perform well on unseen data. The YOLOv11m model separation and DFL losses undergo slight changes but become stable during training. The Validation losses reduce gradually, following the training losses, which indicate the existence of good generalization on unseen data. YOLOv11l model demonstrates an increasing downward trend in training, which represents better localization accuracy. The Classification loss is reduced, and training results are consistent. The changes in the model loss and rising accuracy, recall, and mAP values in the YOLOv11xl model indicate successful model training and validation.

Model Testing

To determine the generalization ability of the proposed model in a real-world situation, a held-out test dataset was considered in Fig. 10 (a), namely crack results. Detection precision and robustness were quantitatively analyzed in Fig. 10 (b) using the standard performance measures in an effort to have an unbiased and dependable measure of the effectiveness of the model.

Existing Models Comparison

The performance of the proposed model was systematically compared in Table 5 with state-of-the-art existing models using an identical dataset and evaluation metrics. This comparative analysis, shown in Table 6, highlights the relative improvements achieved by the proposed approach in terms of accuracy and robustness, demonstrating its effectiveness over existing methods

Discussion

In the proposed framework, the CNN-based preprocessing module is designed to boost such features before the detection process. To detect the abnormal cells properly, the CNN's preprocessing module first enhances such features by ensuring that the initial convolutional layers extract fine edge and texture details, and the subsequent layers suppress the background noise and improve the local contrast on those features. Low spatial resolution is maintained by using small convolutional kernels (3×3), which are vital in preserving the continuity of thin crack patterns.

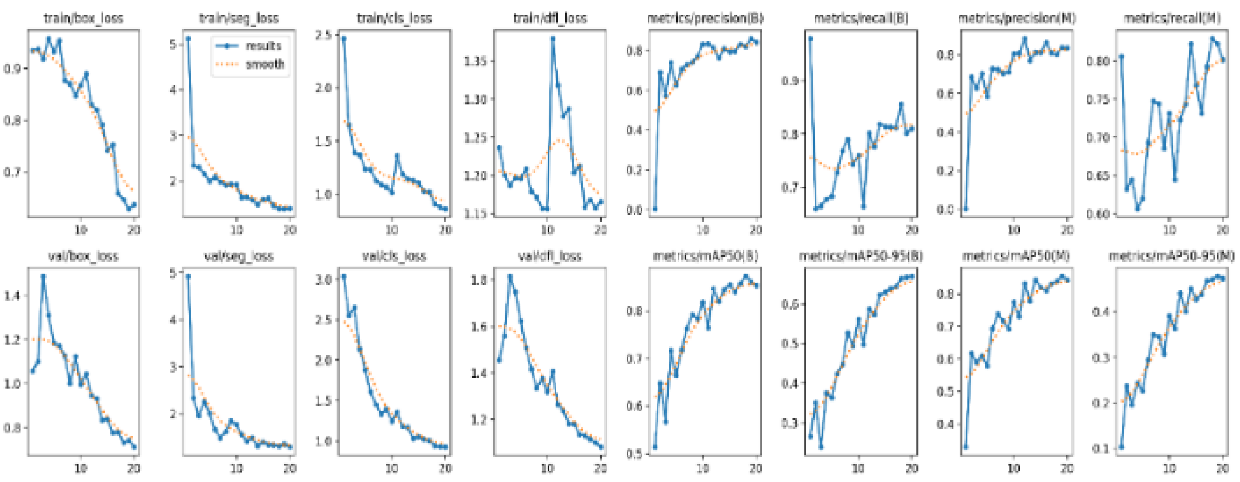

Fig. 9 (i): Metrics Evaluation Graphs YOLOv11n

DOI: 10.3844/jcssp.2026.2411.2424

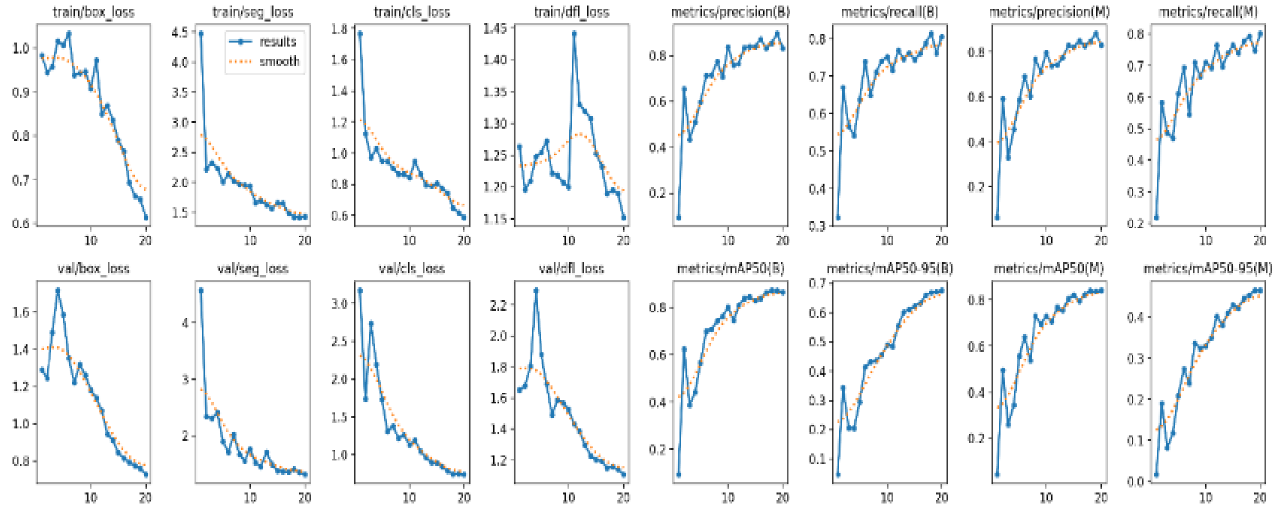

Fig. 9 (ii): Metrics Evaluation Graphs YOLOv11s

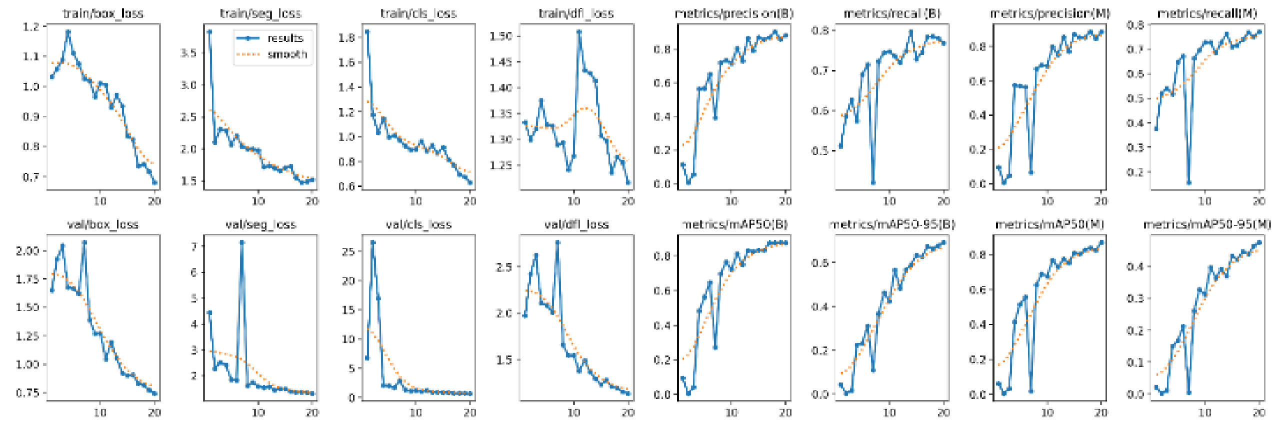

Fig. 9 (iii): Metrics Evaluation Graphs YOLOv11m

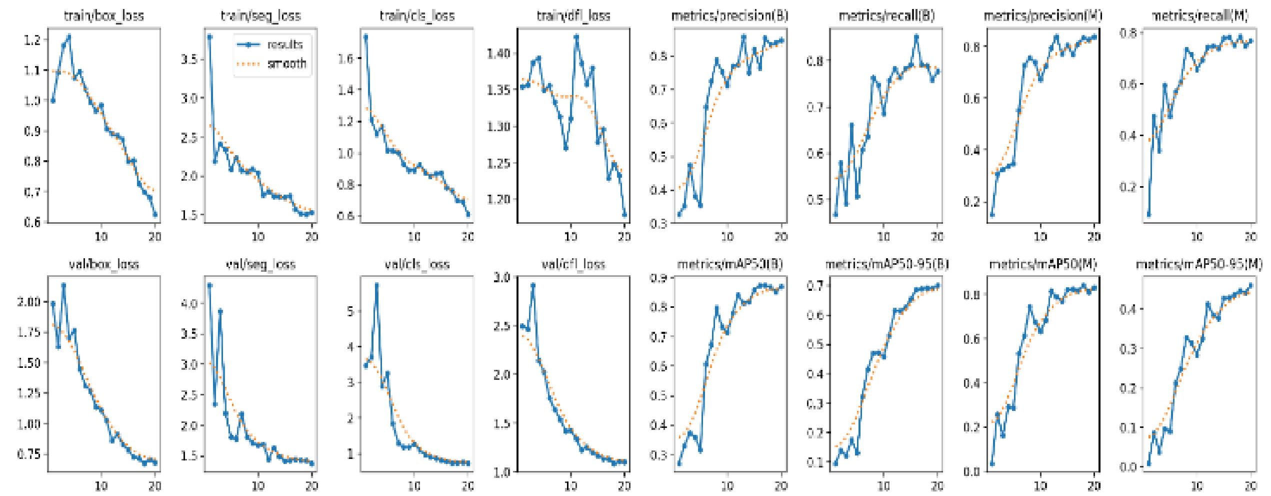

Fig. 9 (iv): Metrics Evaluation Graphs YOLOv11l

DOI: 10.3844/jcssp.2026.2411.2424

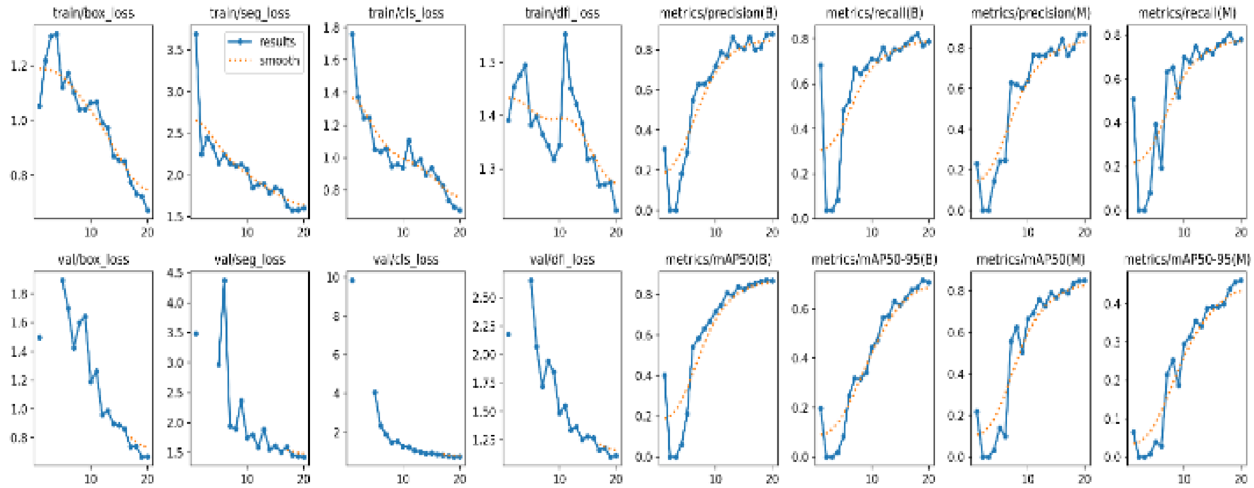

Fig. 9 (v): Metrics Evaluation Graphs YOLOv11xl

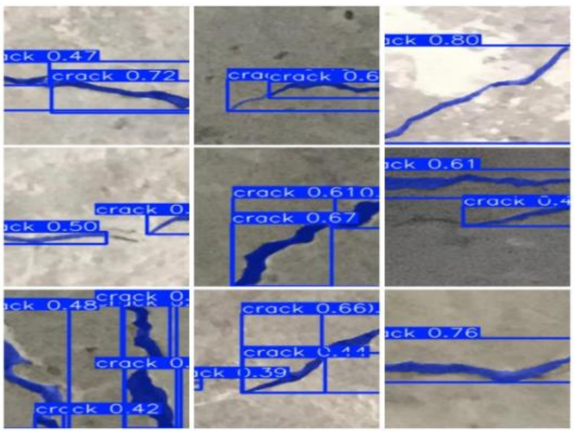

Fig. 10 (a): Crack results

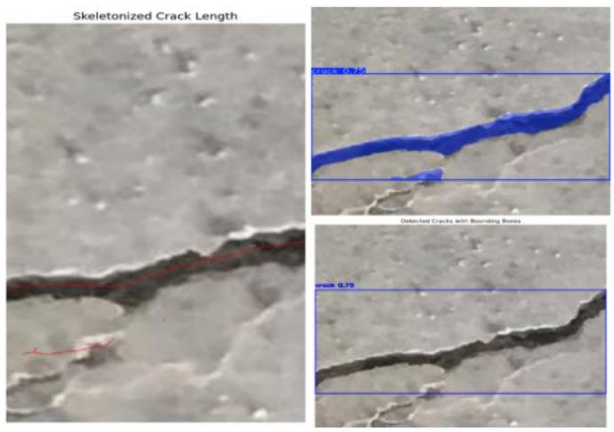

Fig. 10 (b): Calculate Crack length and area

Table 5: Performance Comparison of existing YOLO versions

MODEL	time	Train /box_loss	Train / seg_loss	train/ cls_loss	train/ dfl_loss	metrics /precision(B)	metrics /recall(B)	metrics /mAP50(B)	metrics /mAP50-95(B)	metrics /precision(M)	metrics/ recall(M)	metrics /mAP50(M)	metrics /mAP50-95(M)
YOLO v5s (Pan et al., 2024)	250.00	0.5800	1.2000	0.5300	1.1000	0.8500	0.8200	0.8700	0.6800	0.8300	0.8100	0.8600	0.4700
YOLO v6 (Akyürek et al., 2024)	263.985	0.6375	1.40281	0.8644	1.16574	0.84098	0.80992	0.8538	0.66956	0.83462	0.80165	0.84201	0.47353
YOLO v7 (Karimi et al., 2024)	298.52	0.6136	1.42964	0.58722	1.15209	0.83296	0.80579	0.86468	0.67575	0.82869	0.80165	0.84018	0.46535
YOLO v8 (Wang et al., 2024)	350.0	0.6100	1.3500	0.5500	1.1000	0.8500	0.8200	0.8700	0.6900	0.8400	0.8100	0.8600	0.4700
yolov11 m	505.544	0.68056	1.52158	0.6315	1.21635	0.88304	0.7686	0.87601	0.69254	0.88779	0.77273	0.86967	0.47579

DOI: 10.3844/jcssp.2026.2411.2424

Table 6: Performance Comparison with existing methods

Reference	Year	Authors Name	Technology/Method	Accuracy/Precision
Wang et al. (2019)	2019	Wang, N., Zhao, X., Wang, L., & Zou, Z.	R-CNN	78.2%
Bruno et al. (2023)	2023	S. Bruno, R.A. Galantucci, A. Musicco	Mask R-CNN	0.74- 0.80
Mehta et al. (2023)	2023	S. Mehta, V. Kukreja, A. Gupta	CNN and support vector machine (SVM)	80.2%
Wei et al. (2023)	2023	G. Wei, F. Wan, W. Zhou, C. Xu, Z. Ye, W. Liu, G. Lei, L. Xu	several versions of YOLOv7	77.6%-81.6%
Ravichand et al. (2022)	2022	M. Ravichand, R. Kumar, B. Hazela, T. Suthar	deep CNNs and SVMs	86%
Idjaton et al. (2023)	2023	K. Idjaton, R. Janvier, M. Balawi, X. Desquesnes, X. Brunetaud, S. Treuillet	YOLO network (YOLOv5 incorporating transformer)	79%
Chawla et al. (2023)	2023	T. Chawla, D. Kumar, V. Kukreja	enhanced ET-YOLOV5	88%
Liu et al. (2022)	2022	Z. Liu, R. Brigham, E.R. Long, L. Wilson, A. Frost, S.A. Orr, J. Grau-Bové	DeeplabV3 segmentation models	84.4%
Lin et al. (2024)	2024	T. H. Lin, C. T. Chang, T. H. Zhuang, A. Putranto	AID-Stick (TinyML)	81.25%
Tian et al. (2025)	2025	Z. Tian, F. Yang, L. Yang, Y. Wu, J. Chen, P. Qian	YOLOv11	mAP = 0.607
Proposed Work	2025	Vinod Kumar Pal, Pankaj Mudholkar	Optimized YOLOv11 + Enhanced CNN Pre-processing (Hybrid)	88.77% precision

Conclusion and Future Work

The given research paper dwells upon the use of the YOLOv11 algorithm as well as its evaluation in terms of detecting and categorizing tile defects. Based on the experiments, it was found that YOLOv11 was more accurate, efficient in terms of computing, and had higher precision and recall, as well as mean Average Precision (mAP) when compared to its predecessors. This algorithm has a high detection capability of small and subtle defects, including micro cracks and anomalies on surfaces, higher mAP50-95, and mask-based segmentation. It uses a hybrid architecture, which combines the improvement of transformers with advanced convolutional processing, Which permits the effective extraction of features and detection of defects, even in the presence of adverse conditions, such as low contrast or noise interference. A comparative analysis of the various versions of YOLO indicated that YOLOv11 was characterized by significant advantages regarding training loss measures (box, segmentation, and classification losses) and shorter inference times, thus being a good fit in real-time industrial work. Modern design and capability of supporting multiple defect types and sizes make it easy to integrate into an automated quality control system. These findings underline the fact that YOLOv11 can be used to transform the tile manufacturing sector, as it greatly decreases the expenses incurred in the process of manual inspection and improves the quality of the products by its high detection rate and processing speed.

Based on the positive findings of YOLOv11, further research can focus on several aspects to improve the use of the YOLOv11 model for identifying tile defects. 1 is the expansion of the available dataset to include more images of defects of different types, lighting conditions, and tile models for a more stable model.

Acknowledgment

The authors acknowledge the support provided by their institution in facilitating this research on crack defect detection using the YOLOv11 framework.

Funding Information

No external funding was received for this study.

Author's Contributions

Vinod Kumar Pal: Designed and conceptualized the study, coordinated research activities, collected and analyzed data, conducted the literature review, guided model performance, interpreted the results, and prepared the manuscript.

Pankaj Mudholkar: Critically reviewed the manuscript and provided key insights for the discussion.

Ethics

This research did not involve human participants or animals. The study utilized a publicly available image dataset; therefore, ethical approval was not required.

The authors declare that there is no conflict of interest that could have influenced the conduct or reporting of this research.

References

Abbas, S., Moumen, H., & Abbas, F. (2023). Efficient Method Using Attention Based Convolutional Neural Networks for Ceramic Tiles Defect Classification. *Revue d'Intelligence Artificielle*, *37*(1), 53–62. https://doi.org/10.18280/ria.370108

Akyürek, H. A., Kozan, H. İ., & Taşdemir, Ş. (2024). Surface Crack Detection in Historical Buildings with Deep Learning-based YOLO Algorithms: A Comparative Study. *Computational Research Progress in Applied Science & Engineering*, *10*(3), 1–14. https://doi.org/10.61186/crpase.10.3.2904

Bruno, S., Galantucci, R. A., & Musicco, A. (2023). Decay detection in historic buildings through image-based deep learning. *VITRUVIO - International Journal of Architectural Technology and Sustainability*, *8*, 6–17. https://doi.org/10.4995/vitruvio-ijats.2023.18662

Carvalho, R., Morgado, A. C., Gonçalves, J., Kumar, A., Rolo, A. G. e S., Carreira, R., & Soares, F. (2023). Computer-Aided Visual Inspection of Glass-Coated Tableware Ceramics for Multi-Class Defect Detection. *Applied Sciences*, *13*(21), 11708. https://doi.org/10.3390/app132111708

Chawla, T., Kumar, D., & Kukreja, V. (2023). An Enhanced YOLOV5 Model for Gateways Recognition in Heritage Buildings. *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 736–740. https://doi.org/10.1109/icecaa58104.2023.10212376

Coskun, H., Yiğit, T., Üncü, İ. S., Ersoy, M., & Topal, A. (2022). An Industrial Application Towards Classification and Optimization of Multi-Class Tile Surface Defects Based on Geometric and Wavelet Features. *Traitement Du Signal*, *39*(6), 2011–2022. https://doi.org/10.18280/ts.390613

Cumbajin, E., Rodrigues, N., Costa, P., Miragaia, R., Frazão, L., Costa, N., Fernández-Caballero, A., Carneiro, J., Buruberri, L. H., & Pereira, A. (2023). A Real-Time Automated Defect Detection System for Ceramic Pieces Manufacturing Process Based on Computer Vision with Deep Learning. *Sensors*, *24*(1), 232. https://doi.org/10.3390/s24010232

Dong, X., Yuan, J., & Dai, J. (2025). Study on Lightweight Bridge Crack Detection Algorithm Based on YOLO11. *Sensors*, *25*(11), 3276. https://doi.org/10.3390/s25113276

Gungor, M. A. (2022). A New Gradient Based Surface Defect Detection Method for the Ceramic Tile. *Sakarya University Journal of Science*, *26*(6), 1159–1169. https://doi.org/10.16984/saufenbilder.1038022

Huang, Y., Huang, Z., & Jin, T. (2024). DEU-Net: A Multi-Scale Fusion Staged Network for Magnetic Tile Defect Detection. *Applied Sciences*, *14*(11), 4724. https://doi.org/10.3390/app14114724

Ibrahim, S., Vauxhall, N. A. E. M., Mangshor, N. N. A., Fadzil, A. F. A., & Ghani, N. A. M. (2023). Automated Defective Ceramic Tiles Classification using Image Processing Techniques. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *32*(3), 355–365. https://doi.org/10.37934/araset.32.3.355365

Idjaton, K., Janvier, R., Balawi, M., Desquesnes, X., Brunetaud, X., & Treuillet, S. (2023). Detection of limestone spalling in 3D survey images using deep learning. *Automation in Construction*, *152*, 104919. https://doi.org/10.1016/j.autcon.2023.104919

Karimi, N., Mishra, M., & Lourenço, P. B. (2024). Deep learning-based automated tile defect detection system for Portuguese cultural heritage buildings. *Journal of Cultural Heritage*, *68*, 86–98. https://doi.org/10.1016/j.culher.2024.05.009

Khanam, R., Hussain, M., Hill, R., & Allen, P. (2024). A Comprehensive Review of Convolutional Neural Networks for Defect Detection in Industrial Applications. *IEEE Access*, *12*, 94250–94295. https://doi.org/10.1109/access.2024.3425166

Lin, T.-H., Chang, C.-T., Zhuang, T.-H., & Putranto, A. (2024). Real-time hollow defect detection in tiles using on-device tiny machine learning. *Measurement Science and Technology*, *35*(5), 056006. https://doi.org/10.1088/1361-6501/ad2665

Lu, F., Zhang, Z., Guo, L., Chen, J., Zhu, Y., Yan, K., & Zhou, X. (2022). HFENet: A lightweight hand-crafted feature enhanced CNN for ceramic tile surface defect detection. In *International Journal of Intelligent Systems* (Vol. 37, Issue 12, pp. 10670–10693). https://doi.org/10.1002/int.22935

Liu, Z., Brigham, R., Long, E. R., Wilson, L., Frost, A., Orr, S. A., & Grau-Bové, J. (2022). Semantic segmentation and photogrammetry of crowdsourced images to monitor historic facades. *Heritage Science*, *10*(1), 1–27. https://doi.org/10.1186/s40494-022-00664-y

Mehta, S., Kukreja, V., & Gupta, R. (2023). Empowering Precision Agriculture: Detecting Apple Leaf Diseases and Severity Levels with Federated Learning CNN. *Conference on Intelligent Technologies (CONIT*, 1–6. https://doi.org/10.1109/CONIT59222.2023.10205784.

Pan, H., Li, G., Feng, H., Li, Q., Sun, P., & Ye, S. (2024). Surface defect detection of ceramic disc based on improved YOLOv5s. In *Heliyon* (Vol. 10, Issue 12, p. e33016). https://doi.org/10.1016/j.heliyon.2024.e33016

Ravichand, M., Kumar, R., Hazela, B., & Suthar, T. (2022). Crack on Brick Wall Detection by Computer Vision using Machine Learning. *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, 1017–1020. https://doi.org/10.1109/iceca55336.2022.10009343

Shen, P., Mei, K., Cao, H., Zhao, Y., & Zhang, G. (2025). Lddfsf-Yolo11: A Lightweight Insulator Defect Detection Method Focusing on Small-Sized Features. In *IEEE Access* (Vol. 13, pp. 90273–90292). https://doi.org/10.1109/access.2025.3569970

Stephen, O., Maduh, U. J., & Sain, M. (2021). A Machine Learning Method for Detection of Surface Defects on Ceramic Tiles Using Convolutional Neural Networks. In *Electronics* (Vol. 11, Issue 1, p. 55). https://doi.org/10.3390/electronics11010055

Tian, Z., Yang, F., Yang, L., Wu, Y., Chen, J., & Qian, P. (2025). An Optimized YOLOv11 Framework for the Efficient Multi-Category Defect Detection of Concrete Surface. In *Sensors* (Vol. 25, Issue 5, p. 1291). https://doi.org/10.3390/s25051291

Wang, D., Peng, J., Lan, S., & Fan, W. (2024). CTDD-YOLO: A Lightweight Detection Algorithm for Tiny Defects on Tile Surfaces. In *Electronics* (Vol. 13, Issue 19, p. 3931). https://doi.org/10.3390/electronics13193931

Wang, F., Jiang, X., Han, Y., & Wu, L. (2025). YOLO-LSDI: An Enhanced Algorithm for Steel Surface Defect Detection Using a YOLOv11 Network. In *Electronics* (Vol. 14, Issue 13, p. 2576). https://doi.org/10.3390/electronics14132576

Wang, K., Li, Z., & Wang, X. (2022a). Concatenated Network Fusion Algorithm (CNFA) Based on Deep Learning: Improving the Detection Accuracy of Surface Defects for Ceramic Tile. *Applied Sciences*, *12*(3), 1249. https://doi.org/10.3390/app12031249

Wang, W., Mi, C., Wu, Z., Lu, K., Long, H., Pan, B., Li, D., Zhang, J., Chen, P., & Wang, B. (2022b). A Real-Time Steel Surface Defect Detection Approach With High Accuracy. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–10. https://doi.org/10.1109/tim.2021.3127648

Wang, N., Zhao, X., Wang, L., & Zou, Z. (2019). Novel System for Rapid Investigation and Damage Detection in Cultural Heritage Conservation Based on Deep Learning. *Journal of Infrastructure Systems*, *25*(3), 04019020. https://doi.org/10.1061/(asce)is.1943-555x.0000499

Wei, G., Wan, F., Zhou, W., Xu, C., Ye, Z., Liu, W., Lei, G., & Xu, L. (2023). BFD-YOLO: A YOLOv7-Based Detection Method for Building Façade Defects. *Electronics*, *12*(17), 3612. https://doi.org/10.3390/electronics12173612

Yu, X., Yu, Q., Mu, Q., Hu, Z., & Xie, J. (2023). MCAW-YOLO: An Efficient Detection Model for Ceramic Tile Surface Defects. *Applied Sciences*, *13*(21), 12057. https://doi.org/10.3390/app132112057

Zhou, X., & Tiong, R. L. K. (2024). Defect Detection in Civil Structure Using Deep Learning Method. *Journal of Physics: Conference Series*, *2762*(1), 012068. https://doi.org/10.1088/1742-6596/2762/1/012068

Zhu, W., Wang, Q., Luo, L., Zhang, Y., Lu, Q., Yeh, W.-C., & Liang, J. (2022). CPAM: Cross Patch Attention Module for Complex Texture Tile Block Defect Detection. *Applied Sciences*, *12*(23), 11959. https://doi.org/10.3390/app122311959