Crack detection in structural elements using Haralick Features

Arif Zafar1[0009-0007-9284-5359], Mati ullah2[0000-0002-7212-8826], Afaq Ahmad3[0000-0001-9427-4296],

1 Department of Electrical Engineering, University of Engineering and Technology Taxila,

Pakistan, arif.zafar@students.uettaxila.edu.pk

2Department of Civil and Environmental Engineering, Qatar University, Qatar

matiullah@qu.edu.qa

3Department of Built Environment, Oslo Metropolitan University, Oslo, Norway,

afaq.ahmad@oslomet.no

**Abstract.** Timely detection of cracks in structural elements is essential for ensuring the safety and durability of civil infrastructure. Traditional inspection techniques, such as manual visual assessments, are often labor-intensive, subjective, and ineffective for large-scale or hard-to-access structures. This paper presents an automated crack detection framework based on Haralick texture features extracted from Grey-Level Co-occurrence Matrices (GLCM). The proposed methodology includes image preprocessing, feature extraction, feature selection, and classification. Evaluation is conducted using the publicly available SDNET 2018 dataset, which contains concrete surface images captured under diverse lighting and crack conditions. To enhance computational efficiency, redundant features are eliminated using Ridge and LASSO regression techniques. Support Vector Machines (SVM) with various kernel functions are employed for classification, validated through a 5-fold cross-validation strategy. Experimental results show that the proposed method achieves over 95% accuracy using a subset of selected features, demonstrating its effectiveness and robustness in crack detection for concrete structures.

**Keywords:** Structural Health Monitoring, Crack detection, Haralick features, Grey-Level Co-occurrence Matrix (GLCM), Support Vector Machine (SVM).

1. Introduction

Cracks in structural elements such as bridges, decks, and walls are early indicators of potential structural damage. Detecting and analyzing these cracks accurately is essential to ensure the safety, performance, and longevity of civil infrastructure. Traditionally, visual inspections have been the primary method for detecting cracks. However, these methods are often time-consuming, subjective, and impractical for large-scale or hard-to-access structures [1]. The advancement of image processing and machine learning (ML) techniques offers promising alternatives for automated, objective, and scalable crack detection systems [2]. Recent developments in computer vision have enabled the use of image-based crack detection techniques that leverage both classical and learning-based methods [2, 3].

Classical techniques often involve preprocessing, edge detection, thresholding, and morphological operations to highlight crack patterns [4]. However, these approaches may lack robustness under varying environmental conditions such as lighting, noise, surface texture, and crack width [5]. Learning-based methods have demonstrated significant potential in addressing these limitations. Several studies have explored deep learning methods for the automated detection of cracks in structural elements [6-9]. Ahmed et al. [10] employed a pre-trained ResNet50 model on a dataset of 48,000 pavement images to classify cracks with high accuracy, showing the effectiveness of transfer learning for binary classification of structural defects. Qayyum et al. [11] explored the impact of dataset size on classification accuracy using the InceptionV3 model, categorizing images into diagonal, horizontal, vertical, and uncracked types. The study confirmed that larger datasets improve the model's performance. Shahin et al. [12] proposed a hybrid Visual Transformer (ViT) model for bridge crack inspection. Their custom-built model outperformed in terms of training efficiency, achieving over 99% accuracy with reduced computational cost. Ali et al. [13] evaluated five CNN architectures (custom CNN, VGG-16, VGG-19, ResNet-50, and Inception V3) across eight datasets for crack detection. Chaiyasarn et al. [14] proposed a pixel-level crack detection method by integrating CNN and FCN over 3D texture maps generated from photogrammetry. Their method achieved an accuracy of 99.88% and effectively visualized cracks on large structural surfaces. Hacıefendioğlu and Başağa [15] employed the Faster R-CNN model to detect road surface cracks under diverse lighting and weather conditions. Qayyum et al. [16] integrated CNN and image processing techniques to quantify crack features such as angle, width, endpoint length, and actual path length in concrete images. Using a dataset of 32,000 images, their method achieved low relative errors and demonstrated that actual crack paths are longer than endpoint distances. Hacıefendioğlu and Başağa [15] explored the impact of image preprocessing on VGG16-based CNN crack detection, finding that grayscale models nearly matched the RGB model performance. The study emphasizes the value of preprocessing in reducing data dependency and improving generalization in automated crack detection tasks. While these studies have leveraged deep neural networks for crack detection with impressive results, such methods often entail complex architectures, extensive training data, and substantial computational resources.

In the past few years, there has been a growing interest in texture analysis approaches, particularly the use of Haralick features to improve crack detection reliability. Haralick-based methods treat crack detection as a texture classification problem, rather than explicitly segmenting crack pixels [17]. This global approach can be computationally efficient and effective for automated inspection. Haralick features refer to a set of statistical texture descriptors computed from the Gray Level Co-occurrence Matrices (GLCM) of an image [18]. Common Haralick metrics include contrast, energy (angular second moment), homogeneity, correlation, entropy, and others, characterizing the intensity variation and patterns in the local neighborhood [18]. In crack detection, these features serve as global signatures distinguishing a cracked surface, which typically exhibits high contrast and irregular texture due to crack lines, from an intact surface. Haralick features capture texture information invariant to small geometric changes, making detection more robust to noise, illumination changes, or surface blemishes than simple pixel intensity thresholds [18]. Several studies have demonstrated the effectiveness of Haralick and GLCM-based texture features across various domains of material and structural analysis. Liyuan et al. [19] develop a fuzzy neural network using Haralick features for rail surface anomaly detection, achieving an accuracy of over 91% and highlighting the potential of texture features in safety-critical infrastructure monitoring. Webel et al. [20] introduce a rotation-invariant Haralick-based method for the analysis of SEM images of steel microstructures, enabling reliable phase differentiation under varied imaging conditions. Das and Naskar [18] propose a low-dimensional feature vector for image splicing detection, achieving 95% accuracy and emphasizing computational efficiency. Müller et al. [21] proposed an automated image-based method using Haralick and first-order statistical features extracted from speckle-pattern images to classify cracked vs. uncracked specimens in fracture experiments and achieved 99% classification accuracy. Magalhães Júnior et al. [22] used GLCM-based descriptors to detect cracks in concrete surfaces. Four ML models were trained, i.e, Logistic Regression, MLP, Random Forest, and XGBoost. The XGBoost model yielded the best results with 99.65% accuracy and 99.70% sensitivity.

Existing crack detection methods often lack condition-invariant and robust features, making them less effective under varying environmental conditions. Additionally, the absence of feature selection leads to redundancy, which increases computational cost and reduces classification accuracy. To address these challenges, we propose a compact and efficient framework based on Haralick texture features, which capture spatial relationships of pixel intensities in multiple directions. The method includes image preprocessing, extraction of 14 Haralick features across four orientations, and feature selection using Ridge and LASSO regression to retain only the most statistically independent and significant features. Finally, classification is performed using Support Vector Machines (SVM) with various kernel functions to ensure accurate and reliable crack identification. This design enables accurate and reliable crack detection without the need for high-end graphics processing units (GPUs) or deep learning frameworks.

1. Methodology

The sequential workflow for crack detection using Haralick features is illustrated in Fig 1. The process involves image preprocessing, feature extraction, feature selection, and classification, following a structured pipeline for accurate and reliable crack detection.



**Fig 1.** Proposed Methodology

* 1. Dataset and Preprocessing

The proposed approach was evaluated using the publicly available SDNET 2018 dataset [23], which includes 5,760 concrete surface images (1,920 each from decks, pavements, and walls) captured under varying lighting conditions and crack widths. For each category, 960 images contained cracks, and 960 were uncracked. Only structurally relevant and diagnostically useful images were selected for analysis as shown in Fig *2*.

|  |  |  |
| --- | --- | --- |
| A close up of a white surface  AI-generated content may be incorrect. | A close-up of a white surface  AI-generated content may be incorrect. | A close-up of a white surface  AI-generated content may be incorrect. |
| A picture containing ground  Description automatically generated | A close-up of a crack in a concrete surface  AI-generated content may be incorrect. | A picture containing outdoor  Description automatically generated |
| (a) | (b) | (c) |
| **Fig 2.** SDNET dataset: (a) Deck images (b) Pavement images (c) Wall images |

Each image, originally in RGB format, was converted to grayscale to compute the Grey-Level Co-occurrence Matrix (GLCM) efficiently. Given that each image has an 8-bit intensity depth (i.e., 256 grayscale levels), calculating GLCM across all levels would be computationally intensive and often redundant. Therefore, the grayscale images were quantized to 12 levels to preserve essential crack width information while reducing processing complexity. Following quantization, the GLCM was computed for each image using the reduced 12 grey levels. In practical terms, an RGB image *I(a,b)* of dimensions *N×M×3* was first transformed into a grayscale image of size *N×M*, and then into a 12 × 12 GLCM. This dimensionality reduction hierarchy is illustrated in Fig 3 and is further elaborated upon in the subsequent sections.



**Fig 3**. Hierarchical dimensions used in methodology

* 1. Haralick features Extraction

To characterize the texture of cracks in concrete images, 14 Haralick features were computed from the Grey-Level Co-occurrence Matrix (GLCM) as shown in the Table 1. These features capture the spatial relationship between pixel intensities at a specified direction and offset. The GLCM represents the frequency with which a pixel with intensity value *i* occurs in relation to a pixel with value *j*, separated by a defined offset, typically 1 pixel, in four directions, i.e., 0o,45o,90o, and 135o. Given the sensitivity of pixel offsets to crack widths, this directional analysis helps to preserve structural texture details.

The grayscale images were quantized to *L* levels to manage computational complexity without significantly compromising texture information, and the GLCM was computed accordingly. The resulting matrix *I(i, j)* of size *L×L* captures directional cooccurrence patterns across the quantized levels. For each image, this process yields a total of 56 Haralick features (14 features × 4 directions). These features can be concatenated into a 1 × 56 vector or optionally reduced to a 1 × 14 vector by averaging feature values across all four directions. While the reduced representation lowers computational load, it may slightly decrease accuracy and limit the ability to localize crack orientation, as directional extremities such as local maxima or minima carry meaningful structural information. The impact of crack orientation on feature behavior is further analyzed in the results section.

* 1. Feature Selection

Certain Haralick features exhibit linear dependencies, leading to redundancy that can negatively impact model performance and increase computational overhead. To address this, feature selection is essential to retain only the most informative and independent features. We adopted embedded feature selection methods that integrate model training with feature evaluation. Specifically, LASSO (Least Absolute Shrinkage and Selection Operator) and Ridge Regression were employed to rank features based on their significance. These methods apply a regularization constraint that penalizes features with lower importance by shrinking their corresponding weights. When a feature's weight exceeds the specified threshold, it is penalized more heavily, thereby reducing its influence in the final model.

**Table 1.** Selected Haralick Features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Formula |  | Feature | Formula |
| *Contrast (cont.)* | $$=\sum\_{i,j}^{N-1}(i-j)^{2}\*I(i,j)$$ |  | *Sum average (SA)* | $$=\sum\_{i=2}^{2N\_{g}}i I\_{(x+y)}(i)$$ |
| *ASM* | $$=\sum\_{i,j}^{N-1}[I(i,j)]^{2}$$ |  | *Sum variance (SV)* | $$=\sum\_{i=2}^{2N\_{g}}(i-SE)^{2}I\_{(x+y)}(i)$$ |
| *Energy (E)* | $$=\sqrt{ASM}$$ |  | *Sum Entropy (SE)* | $$=\sum\_{i=2}^{2N\_{g}}I\_{(x+y)}(i)×log\_{I\_{(x+y)}(i)}$$ |
| *Homogeneity/Maximal correlation coefficient (Hom/MCC)* | $$=\sum\_{i,j}^{}\frac{I(i,j)}{1+\left|i-j\right|}$$ |  | *Difference Variance (DV)* | $$=\sum\_{i=0}^{N\_{g}-1}i^{2}I\_{(x-y)}(i)$$ |
| *Correlation (Corr)* | $$=\sum\_{i,j}^{N-1}\frac{\left(i,j\right)\*I\left(i,j\right)-μ\_{x}\*μ\_{y}}{σ\_{x}\*σ\_{y}}$$ |  | *Difference Entropy (DE)* | $$=\sum\_{i=2}^{N\_{g}-1}I\_{(x-y)}(i)×log\_{I\_{(x-y)}(i)}$$ |
| *Variance (V)* | $$=\sum\_{i,j}^{N-1}(i-μ)^{2}\*I(i,j)$$ |  | *IMC1* | $$\frac{Entropy-HXY1}{max(HX,HY)}$$ |
| *Inverse difference moment (IDM)* | $$=\frac{∑\_{i}∑\_{j} I\_{(i,j)}}{1+(i-j)^{2}}$$ |  | *IMC2* | $$=(1-exp(-2(HXY2-Entropy)))^{0.5}$$ |

* 1. Classifier

Support Vector Machines (SVMs) were employed to effectively distinguish between crack and non-crack image classes. SVMs work by projecting input data into a higher-dimensional space where an optimal hyperplane can be constructed to separate linearly or non-linearly separable classes. In this study, we utilized polynomial and Radial Basis Function (RBF) kernels to handle the nonlinear relationships inherent in texture-based features. Model performance was evaluated using a k-fold cross-validation scheme to ensure robust training and testing across the dataset and to minimize overfitting

1. Results and discussion

The performance of the proposed crack classification model was evaluated using standard metrics: Accuracy (Acc), Recall (R), Precision (Pre), and F1-score. These metrics were derived from the confusion matrix, which comprises four values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Table 2 presents the classification results, with the feature length (i.e., the number of Haralick features used) indicated in the metrics column. Various SVM kernels, including Quadratic (Q), Cubic (C), and Fine Gaussian (FG), were tested across three image categories: Deck, Pavement, and Wall.

Among the tested kernels, the Cubic polynomial kernel consistently yielded superior performance across most categories. Notably, when the feature vector was reduced from 56 to 14 dimensions by averaging Haralick features across directions, the model's performance remained largely unaffected, demonstrating the efficacy of dimensionality reduction without significant loss of classification accuracy.

**Table 2.** Performance metrics for SVM kernels with 5-Cross-Fold validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metrics | All | Deck | Pavement | Wall |
| Kernel | Q | C | FG | Q | C | FG | Q | C | FG | Q | C | FG |
| Acc(56) | 93 | 94 | 92 | 92 | 91 | 89 | 98 | 98 | 96 | 95 | 96 | 93 |
| Acc(14) | 92 | 93 | 89 | 90 | 90 | 87 | 97 | 97 | 94 | 94 | 94 | 92 |
| R(14) | 94 | 95 | 92 | 94 | 93 | 91 | 98 | 98 | 96 | 96 | 96 | 95 |
| Pre (14) | 89 | 90 | 87 | 87 | 87 | 85 | 95 | 95 | 94 | 91 | 92 | 89 |
| F1(14) | 91 | 92 | 89 | 90 | 90 | 88 | 97 | 97 | 94 | 95 | 96 | 93 |

* 1. Individual performance of Haralick Features

To assess the discriminative power of each Haralick feature independently, classification experiments were conducted using individual features as input to the SVM classifier. The performance was evaluated across various kernel types, including Fine Gaussian (FG) and Medium Gaussian (MG) as shown in Fig 4.

**Fig 4.** Individual performance of haralick features over SVM kernels

The results indicate that certain features, such as Variance and Sum Variance, consistently outperform others, particularly when used with Gaussian-based kernels. These features demonstrate a strong ability to capture texture variations associated with cracks, making them highly effective for classification. In contrast, features like Coarse Gaussian (CG) and Linear Polynomial (L) showed limited performance, suggesting lower discriminative capability when used in isolation.

* 1. Feature Selection using Ridge and LASSO Methods

To enhance classification efficiency and eliminate feature redundancy, embedded feature selection techniques were employed specifically, Ridge Regression and LASSO. These methods rank features based on their contribution to the model's predictive performance while applying regularization to suppress less significant ones.

In the case of Ridge Regression, the top six features, Contrast, Sum Variance (SV), Angular Second Moment (ASM), Difference Entropy (DE), IMC2, and Sum Entropy (SE), were identified as the most impactful. As shown in Table 3, the Quadratic kernel achieved the highest classification accuracy of 88.6% with these features, using a regularization parameter of 4.094809 and a constraint of 0.000001.

**Table 3.** Feature Selection Using Ridge Regression On All Images

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SVM kenel | L | Q | C | FG | MG | CG |
| Accuracy  | 72.5 | *88.6* | 57.4 | 85.6 | 84.8 | 75.2 |

Similarly, LASSO regression selected features such as Maximal Correlation Coefficient (MCC), IMC2, IMC1, Difference Entropy (DE), Difference Variance (DV), and Energy (E) also demonstrated comparable performance, as shown in Table 4, with a maximum accuracy of 87.2% under the quadratic kernel, using a regularization parameter of 0.0001 and a constraint of 100000. These results confirm the effectiveness of both Ridge and LASSO in identifying a compact and informative feature subset, significantly reducing dimensionality without compromising classification performance.

**Table 4.** Feature selection using LASSO regression on all images

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SVM kernel** | **L** | **Q** | **C** | **FG** | **MG** | **CG** |
| Accuracy  | 67.1 | 87.2 | 52.7 | 86.3 | 85.8 | 74.5 |

Feature dependencies were further analyzed by computing the correlation matrix among all Haralick features, as visualized in the heatmap shown in Fig 5. In this matrix, diagonal elements represent self-similarity and thus have a value of 1, which is expected. Features with a correlation coefficient greater than 0.5 are considered highly dependent and appear as brighter regions in the heatmap. In contrast, independent features exhibit correlation values closer to zero and are represented by darker regions. This visual analysis provides valuable insight into feature redundancy and supports the feature selection process by identifying which features contribute unique information.



**Fig 5.** Correlation Heatmap Between Haralick features

* 1. CLAHE Preprocessing

Enhancing the contrast between crack and non-crack regions is intuitively expected to improve classification accuracy. However, traditional methods such as global histogram equalization and adaptive histogram equalization have limitations. The former assumes a uniform distribution of pixel intensities, making it ineffective for non-uniform textures, while the latter enhances local contrast but can also amplify noise, particularly in homogeneous regions. To address these issues, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied. As shown in Table 5, the classification accuracy of SVM kernels after applying CLAHE preprocessing varies across image categories. Notably, the Pavement category achieved the highest accuracy (98%) with both Quadratic and Cubic kernels, whereas performance slightly declined for the Deck category, with the Cubic kernel (87%). This indicates that while CLAHE enhances contrast, it may also introduce texture artifacts in certain cases, potentially affecting the model's performance.

**Table 5.** SVM kernels accuracy after applying CLAHE as pre-processing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | All | Deck | Pav | Wall |
| kernel | Q | C | F | Q | C | F | Q | C | F | Q | C | F |
| Acc | 92 | 93 | 90 | 90 | 87 | 90 | 98 | 98 | 95 | 95 | 95 | 92 |

CLAHE enhances local contrast within defined regions while limiting the amplification of noise by clipping the histogram at a predefined threshold. As illustrated in Fig 6, CLAHE successfully enhances fine details in the image, making crack textures more prominent. However, this enhancement also affects non-crack images, often transforming smooth textures into coarse ones. This unintended effect can introduce false patterns, leading to a reduction in classification accuracy, particularly when using the full 1×56 Haralick feature set, as evidenced by the results in Table 5.

|  |  |
| --- | --- |
| A crack in a wall  AI-generated content may be incorrect. | A crack in a wall  AI-generated content may be incorrect. |
| A close-up of a white wall  AI-generated content may be incorrect. | A close-up of a grey wall  AI-generated content may be incorrect. |
| (a) | (b) |

**Fig 6.** CLAHE (a) Before preprocessing (b) After the preprocessing

* 1. Crack Localization

An important observation from the feature analysis is that certain Haralick features, when considered independently, exhibit local extrema aligned with the direction of the crack. For instance, the contrast feature tends to reach local maxima in the direction of the crack due to significant intensity differences between adjacent pixel neighborhoods.

In crack-containing images, the normalized difference between the local maximum and the corresponding feature values in other directions often exceeds 50%, indicating a strong directional response corresponding to the crack path. In contrast, for non-crack images, this difference typically remains below 50%, unless the image contains surface irregularities such as texture deformities or bugholes, which may produce false extrema. Table 6 represents the contrast feature values in four directions $\left(0^{∘},45^{∘},90^{∘},135^{∘}\right)$ respectively rounded to two decimal places for both crack and non-crack image. The dissimilarity metric, defined in Equation (1), captures the normalized deviation of values from the local maximum across directions. This analysis supports the potential of Haralick features not only for classification but also for localizing cracks based on directional texture variations.

$Dissimilarity (\%)=\frac{\left‖X\_{min}-X\_{i}\right‖}{X\_{min}}×100$ ( 1 )

**Table 6.** Contrast feature values and dissimilarity between contrast and local maxima for crack & non-crack images

|  |  |  |
| --- | --- | --- |
| Category | Crack | Non-Crack |
| Contrast | 0.58 | 0.53 | 0.42 | 1.13 | 1.66 | 2.22 | 1.02 | 2.46 |
| Dissimilarity | 55 | 59 | 71 | 0 | 32 | 9 | 58 | 0 |

# Conclusion

This study presented a computationally efficient and accurate framework for automated crack detection in structural elements using Haralick texture features derived from Grey-Level Co-occurrence Matrices (GLCM). Unlike deep learning-based methods that often demand large training datasets and high-end computational resources, the proposed approach leverages handcrafted texture descriptors, feature selection techniques, and Support Vector Machine (SVM) classifiers to achieve high detection performance. Experimental results on the SDNET 2018 dataset demonstrate that the method achieves over 95% accuracy using a reduced set of features, with certain individual Haralick features, such as Contrast and Sum Variance, showing strong discriminative power. Feature selection through Ridge and LASSO regression not only reduced redundancy but also improved model interpretability and efficiency. Moreover, the application of CLAHE preprocessing and directional analysis provided further insights into crack localization, highlighting the proposed method’s potential beyond binary classification. The proposed framework provides a robust and lightweight alternative for condition-invariant crack detection, serving as a practical solution for real-time structural health monitoring applications, particularly in scenarios where computational resources are limited.

**Acknowledgments**

The authors are thankful to Oslomet University for investigating this research.

**Funding**

This is the part of the Project “Forensic Analysis of Concrete Through Image Processing” FACIP, funded by the EU against

The Grant agreement ID: 101153307, <https://cordis.europa.eu/project/id/101153307>.

**References**

[1] Mohammed, M., *Bridge Inspection Using Drone & AI*. 2025, Purdue University Graduate School.

[2] Ullah, M., J. Mir, S.S. Husain, M.L.U.R. Shahid, and A. Ahmad, *Concrete forensic analysis using deep learning-based coarse aggregate segmentation.* Automation in Construction, 2024. **162**: p. 105372.

[3] Ma, N., et al., *Vehicular Road Crack Detection with Deep Learning: A New Online Benchmark for Comprehensive Evaluation of Existing Algorithms.* arXiv preprint arXiv:2503.18082, 2025.

[4] Al-Mistarehi, B., *An approach for automated detection and classification of pavement cracks*. 2017.

[5] Ahmad, C.F., et al., *Classification of potholes based on surface area using pre-trained models of convolutional neural network.* arXiv preprint arXiv:2309.17426, 2023.

[6] Laxman, K., N. Tabassum, L. Ai, C. Cole, and P. Ziehl, *Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning.* Construction and Building Materials, 2023. **370**: p. 130709.

[7] Hacıefendioğlu, K., A.C. Altunışık, and T. Abdioğlu, *Deep learning-based automated detection of cracks in historical masonry structures.* Buildings, 2023. **13**(12): p. 3113.

[8] Park, S.E., S.-H. Eem, and H. Jeon, *Concrete crack detection and quantification using deep learning and structured light.* Construction and Building Materials, 2020. **252**: p. 119096.

[9] Ehtisham, R., W. Qayyum, V. Plevris, J. Mir, and A. Ahmad. *Classification and computing the defected area of knots in wooden structures using image processing and CNN*. in Proceedings of the *Proceedings of the 5th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control (EUROGEN 2023), Chania, Crete, Greece*. 2023. p. 1-3.

[10] Ahmed, C.F., A. Cheema, W. Qayyum, and R. Ehtisham, *Detection of pavement cracks of UET Taxila using pre-trained model Resnet50 of CNN.* IEEE Access, 2022. **7**: p. 176065-176086.

[11] Qayyum, W., R. Ehtisham, C. Camp, J. Mir, and A. Ahmad. *Detecting cracks with Convolution Neural Network (CNN) with Variable image dataset*. in Proceedings of the *2nd International Conference on Recent Advances in Civil Engineering and Disaster Management*. 2022. Peshawar Pakistan. p. 166-170.

[12] Shahin, M., et al., *Improving the concrete crack detection process via a hybrid visual transformer algorithm.* Sensors, 2024. **24**(10): p. 3247.

[13] Ali, L., et al., *Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures.* Sensors, 2021. **21**(5): p. 1688.

[14] Chaiyasarn, K., et al., *Integrated pixel-level CNN-FCN crack detection via photogrammetric 3D texture mapping of concrete structures.* Automation in Construction, 2022. **140**: p. 104388.

[15] Hacıefendioğlu, K. and H.B. Başağa, *Concrete road crack detection using deep learning-based faster R-CNN method.* Iranian Journal of Science and Technology, Transactions of Civil Engineering, 2022. **46**(2): p. 1621-1633.

[16] Qayyum, W., et al., *Predicting characteristics of cracks in concrete structure using convolutional neural network and image processing.* Frontiers in Materials, 2023. **10**: p. 1210543.

[17] Löfstedt, T., P. Brynolfsson, T. Asklund, T. Nyholm, and A. Garpebring, *Gray-level invariant Haralick texture features.* PloS one, 2019. **14**(2): p. e0212110.

[18] Das, D. and R. Naskar, *Image splicing detection using low-dimensional feature vector of texture features and Haralick features based on Gray Level Co-occurrence Matrix.* Signal Processing: Image Communication, 2024. **125**: p. 117134.

[19] Liyuan, Y., Y. Ming, G. Osman, X. Juan, and Y. Shishuai. *Anomaly Detection on Railway Track Using Haralick Feature for Color Mapping Co-Occurrence Matrix*. in Proceedings of the *2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*. 2024. IEEE. p. 301-306.

[20] Webel, J., J. Gola, D. Britz, and F. Mücklich, *A new analysis approach based on Haralick texture features for the characterization of microstructure on the example of low-alloy steels.* materials Characterization, 2018. **144**: p. 584-596.

[21] Müller, A., N. Karathanasopoulos, C.C. Roth, and D. Mohr, *Machine learning classifiers for surface crack detection in fracture experiments.* International Journal of Mechanical Sciences, 2021. **209**: p. 106698.

[22] Magalhães Júnior, A., Í. Sousa, H. Sousa, T. Sáfadi, and P. Guimarães, *Cracks Detection in Concrete Surfaces using Machine Learning.*

[23] Dorafshan, S., R.J. Thomas, and M. Maguire, *SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks.* Data in brief, 2018. **21**: p. 1664-1668.