WASD
WORLD ASSOCIATION FOR SUSTAINABLE DEVELOPMENT

W | J | S | T | S | D
World Journal of Science, Technology and Sustainable Development

ISSN: 2042-5945 (Print) | 2042-5953 (Online)
WJSTSD V20 N3 2025

OPEN ACCESS
DOI: 10.47556/J.WJSTSD.20.3.2025.5

RESEARCH

Performance Evaluation of AI and Traditional Techniques for Crack Detection on Concrete Structures

Omer Mohamed Omer Abdalla
EMJMD in Mechatronics (EU4M) University of Oviedo
Oviedo, Spain
Email: Omer.abdalla@eu4m.eu
ORCID: 0009-0000-5627-7453

Ángel Encalada-Dávila
EMJMD in Mechatronics (EU4M) University of Oviedo
Oviedo, Spain
Email: angel.encalada@eu4m.eu
ORCID: 0000-0001-6259-8464

Mitterand Sani Ekole
EMJMD in Mechatronics (EU4M) University of Oviedo
Oviedo, Spain
Email: mitterand.ekole@eu4m.eu
ORCID: 0000-0002-3250-7100

ABSTRACT

PURPOSE: This study develops and compares AI-based and traditional crack detection models for concrete structures to enhance timely maintenance and prevent costly repairs.

DESIGN/METHODOLOGY/APPROACH: A Convolutional Neural Network (ResNet50) with dynamic quantisation (DQ) is used for reduced memory, faster inference and lower energy consumption. A traditional model based on histogram intersection is also employed. Both models are trained and tested on publicly available datasets of concrete structures and compared in accuracy, precision, recall and training time.

CITATION: Abdalla, O. M. O., Encalada-Dávila, Á and Ekole, M. S. (2025): Performance Evaluation of AI and Traditional Techniques for Crack Detection on Concrete Structures. *World Journal of Science, Technology and Sustainable Development,* Vol. 20, No. 3, pp.281-295.

RECEIVED: 19 February 2025 / **REVISED:** 15 July 2025 / **ACCEPTED:** 15 August 2025 / **PUBLISHED**: 30 September 2025

FINDINGS: The DQ-ResNet50 model shows training accuracy from 0.780 to 0.917 and validation accuracy from 0.707 to 0.906, while the traditional model's accuracy drops to 0.630 in validation. The DQ-ResNet50 offers better deployment efficiency.

VALUE: The DQ-ResNet50 model balances performance and deployment benefits, making it more suitable for real-world applications.

KEYWORDS: *Crack Detection; Concrete Structures; Resnet50; Dynamic Quantisation; Traditional Techniques; Histogram Intersection.*

INTRODUCTION

Concrete structures are essential for modern infrastructure, but cracks can compromise their integrity and pose safety hazards. Early crack detection is crucial for timely repair and maintenance, which can prevent costly repairs or replacement in the future. Early detection can also reduce the environmental impact of repairing or replacing damaged structures. Visual inspection plays a critical role in the process of assessing the safety and functionality of the concrete infrastructure. However, the reliance on human inspection introduces the potential for errors and uncertainties. There is, therefore, a need for the integration of robotics into Structural Health Monitoring (SHM) frameworks (Koch *et al.*, 2015).

Over the last 20 years, a significant number of crack detection methodologies have been developed. Notably, recent years have witnessed remarkable advancements in the field of crack detection, particularly with the use of deep learning (DL) techniques. These advancements have led to improved accuracy and efficiency in detecting cracks, and have the potential to enhance the safety and reliability of various structures. Over time, image processing algorithms have significantly improved and led to the proposal of many methods that can be broadly categorised into two approaches: classical/traditional and AI-based methods.

Classical Computer Vision (CV) techniques use a novel approach for detecting cracks in concrete structures using basic image processing to extract crack features from images of concrete surfaces. Many research papers proposed the use of classical methodologies such as the integrated algorithms that pre-process the image and then perform the segmentation to isolate cracks (Yu *et al.*, 2007). Another classical technique used is the morphological approach that uses mathematical operations, such as erosion, dilation, opening and closing (Wang *et al.*, 2019). Iyer and Sinha developed a method to detect cracks in contrast-enhanced sewer pipe images by using a curvature feature evaluation with mathematical morphology processing; they were able to measure the crack properties and determine the pipe severity level.

Percolation-based methods involve analysing the connectivity of pixels in an image to identify clusters or regions of interest (Qu *et al.*, 2019). One good demonstration of the viability of this technique is the crack detection approach that prioritised the linearity and continuity of cracks by utilising a grey-scale Hough transform and percolation processing (Singh *et al.*, 2023). The practical technique is a semi-automated technique that is based on the route-finder algorithm whose purpose is to delineate the crack as a polyline (Dare *et al.*, 2002). The scope of classical techniques is wide and diverse; several research papers summarise and review these methods (Ai *et al.*, 2023). Although the classical methods are computationally efficient compared to the AI-based technique, they lack the ability to detect complex cracks and need some level of experience in the domain to be able to interpret their results. The result of classical methods is also affected by the variations of cracks in their shape and magnitude, as well as the varying degrees of cracking. Additionally, the presence of an irregularly patterned background, camera distance and environmental conditions can also affect the accuracy of crack detection (Yamane and Chun, 2020).

CV techniques are being popularly used to analyse cracks and other surface defects in civil infrastructure, utilising different classic and AI-based CV methods (Islam and Kim, 2019; Khan *et al.*, 2023). With the ability to learn from large datasets, AI techniques have shown promising results in classification and detecting cracks in concrete structures, even when dealing with complex and diverse crack patterns. One good example of this is concrete crack detection and localisation using U-Net that was able to identify the crack locations from the input raw images under various conditions (Liu *et al.*, 2019). Another example is the study conducted by Dung and Anh (2019), where a visual geometric group (VGG) was utilised to classify the presence of cracks within a small region; subsequently, a fully convolutional network (FCN) was employed to detect crack pixels individually. Other algorithms have been developed for the purpose of crack detection, including CrackNet (Zhang *et al.*, 2017), as well as improved versions called CrackNet II (Zhang *et al.*, 2018) and Crack-Net-V (Hu *et al.*, 2021).

AI-based techniques have demonstrated high efficiency in crack detection, but this comes at the cost of increased computational power and hardware requirements. Additionally, one of the limitations of using AI-based techniques is the need for extensive datasets to train the models, which may not always be available for certain applications.

In this work, the proposal is based on the following contributions:

- a CNN (ResNet50), based on dynamic quantisation (DQ), that offers a reduced memory footprint, improved inference speed, lower energy consumption, compatibility with hardware acceleration and easier model deployment;
- an image classification model based on a histogram inter-section (traditional technique) that focuses on the image colour distribution to decide if any input image is similar to any of the model's reference images;
- a comparison of these models with previous work, in terms of classification performance metrics such as training time, accuracy, precision and recall.

The remainder of the paper is distributed as follows: the next section discusses data collection and data split, followed by an explanation of the main foundations for the classical and AI-based techniques used in this work.

The penultimate section details the obtained results and discusses them together with a previous work comparison, while the final section gives the essential findings and remarks about this paper.

DATA

In this section, data collection and data split are explained as part of data pre-processing, before using the data to implement any algorithms.

Data Collection

The dataset used in this work was created from two publicly available datasets (Dorafshan *et al.*, 2018; Özgenel and Sorguç, 2018) where the heterogeneity provides a significant effect on the generalisation of DL models. Different datasets were created by varying the size, in order to measure the performance of trained models on dataset- size variation, i.e., 2.8k, 5.6k, 8.4k, 10.4k and 13.4k RGB images with 224x224 resolution. The images correspond to different types of concrete structures, for example, pavements, walls, bridge decks, etc., where there are two possible binary classes: crack (C) (see Figure 1) and non-crack (NC) (see Figure 2).

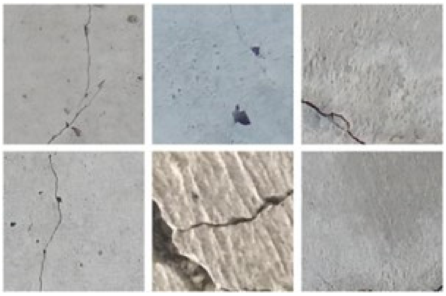

Figure 1 Graphical Examples of Cracks on Concrete Structures

Source: Dorafshan *et al.*, 2018; Özgenel and Sorguç, 2018

Figure 2 Graphical Examples of Non-Cracks on Concrete Structures

Source: Dorafshan *et al.*, 2018; Özgenel and Sorguç, 2018

Data Split

In ML, data split is typically done to avoid overfitting, that is, when an ML model fits its training data too well and fails to reliably fit additional data. The split ratio used in this work is 60% for training, 20% for testing and 20% for validation. The split is stratified randomly, as there are binary classes and two different datasets, so the balance of equally acquiring knowledge from both sources should be guaranteed. Table 1 details the data split by considering the different built datasets and the three subfolders.

Table 1 New Kuwait Vision 2035 vs SDGs

Dataset	Training		Validation		Testing	
	C	NC	C	NC	C	NC
2.8 k	840	840	280	280	280	280
5.6 k	1,680	1,680	560	560	560	560
8.4 k	2,520	2,520	840	840	840	840
10.4 k	3,120	3,120	1,040	1,040	1,040	1,040
13.4 k	4,020	4,020	1,340	1,340	1,340	1,340

Source: Constructed by authors

CLASSIFICATION METHODOLOGY

In this section, the main foundations for AI and traditional techniques are stated in order to have a brief background about how they work and the performance key points.

ResNet50 based on DQ

ResNet is a convolutional neural network (CNN) architecture that was first introduced in the paper "Deep Residual Learning for Image Recognition" (He *et al.*, 2015). It is a deep CNN, with 50 convolutional layers, and has been shown to be very effective for a variety of image classification tasks.

The ResNet architecture is based on the idea of residual learning. Residual learning refers to a way of training CNNs that allows them to learn long-range dependencies in their input data. This is done by adding a shortcut connection to each convolutional layer. The shortcut connection allows the network to learn the residual between the input and output of the convolutional layer. This helps the network to learn more complex features from its input data. ResNet50 is a variant of the ResNet model that has 48 convolution layers along with 1 max-pooling layer and 1 average-pool layer. It has 3.8x109 floating points (FP32) operations. The ResNet50 architecture includes a 7x7 kernel convolution alongside 64 other kernels with a 2-sized stride, a max-pooling layer with a 2-sized stride, and many more layers.

The implementation of ResNet50 in this study was based on DQ (see Figure 3). It relates to methods for computing and storing tensors at smaller bandwidths than FP32 precisions. DL efficacy in image processing and natural language processing tasks has resulted in the creation of an increasing number of applications that execute deep learning models on a variety of platforms, ranging from cloud-scale clusters to resource-limited edge devices (Vaswani *et al.*, 2017; Dosovitskiy *et al.*, 2021). Frameworks such as Pytorch and Tensorflow allow developers and researchers to create and train large-scale models with great processing power. This is because the aforementioned frameworks have a collection of operators, and these operators perform complex mathematical operations.

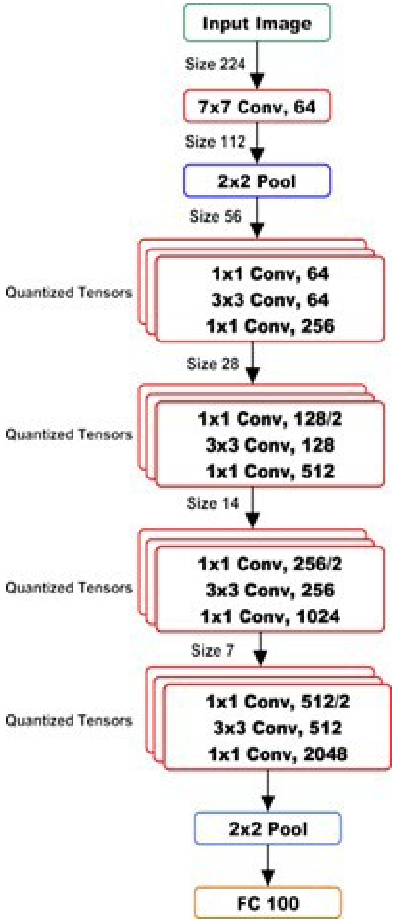

Figure 3 DQ-ResNet50 Architecture
Source: Created by the authors

These operators are often translated to machine code using either NVIDIA's CUDNN (CUDA Deep Neural Network) or Intel's Deep Neural Network Library.

Most DL models constructed with these frameworks define their model's weight and associated activation in terms of FP32; this has a larger bandwidth and is capable of storing large-scale tensors. The disadvantage of adopting FP32 models is the limitation experienced when executing these models on edge devices or low-computation hardware. Quantisation enables the representation of model weights and activations from FP32 into int8 (8-bit integers), resulting in a four-fold reduction in memory footprint (Wu *et al.*, 2020).

The strategy used in this work involved performing a per-channel symmetric quantisation; this means that the values in the tensor are quantised with different quantisation parameters for each channel dimension of the tensor. This allows for less error in converting tensors to quantised values because outliers only affect the channel they were in, rather than the entire tensor. Quantisation-aware training was employed to achieve efficiency and minimal sacrifice on model correctness and efficiency. The main goal of this training procedure is to determine the scale factor that will facilitate dynamic conversion from FP32 to int8 based on the data range observed during training, and to ensure that this scale factor is tuned so that as much information about each observed dataset as possible is preserved after each epoch of training until it is completed. The mapping from FP32 to int8 is represented by:

1. $A_{\mathrm{FP32}5\mathit{scale}\mathrm{A}3(}\mathrm{Q}_{\mathrm{A}2}zp_{\mathrm{A})}$,

where A_{FP32} is the FP32 tensor and Q_A is the int8 tensor, zp_A is a standard zero-point parameter required to ensure mapping. Then, by dynamically quantising (pre-training quantisation) the architecture of ResNet50, the stated model in this work is capable of learning quantisation parameters during training and storing its weights and activations in int8 (8-bit integer) as opposed to FP32. This ensures the overall model size is shrunk by more than 50% and reduces the computational requirements, making it easily deployable on most edge devices.

Histogram-based Colour Classification

Identifying objects through colours only could be considered very challenging at first, however, CNNs, due to their strong processing and understanding capability, can deal with that very easily. Nevertheless, more classical strategies, such as histogram-intersection algorithms, also use colour information to recognise objects (see Figure 4). In their article, Swain and Ballard (1991) proposed the main foundations for such an algorithm. When the colour is a strong predictor of the object's identity this technique is very reliable and useful. Basically, a histogram is a graphical representation of the value distribution of a digital image. That value distribution is a way of representing the colour appearance and in the hue, saturation, value (HSV) model it represents the saturation of a colour. The intersection of two histograms could be computed as:

$$I = \frac{\sum_{i=1}^{n} \min(H_i^a, H_i^b)}{\sum_{i=1}^{n} H_i^b}$$

where the min function takes two-pixel values and returns the smallest one, Hai, is the input-image histogram, Hbi, is the model-image histogram and n is the total number of histogram bins. So, the result of the intersection is the number of pixels from the model image that have matched pixels of the same colours in the input image.

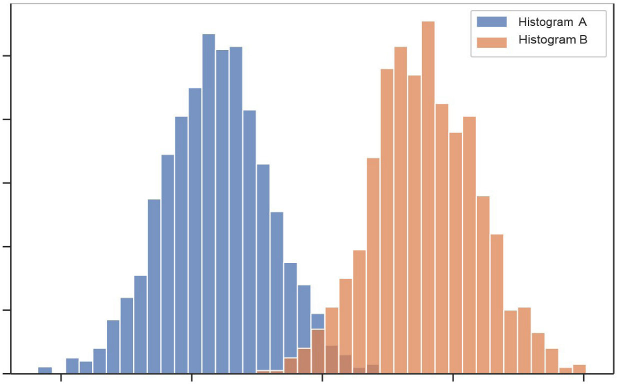

Figure 4 Graphical Representation of Histogram Intersection'

Notes: For the example, both follow a Gaussian distribution, where Histogram A is μ=-4 and σ=2, and Histogram B is μ=4 and σ=2

Source: Created by the authors

RESULTS AND DISCUSSION

In this section, the obtained results from the AI and traditional techniques are contrasted with each other and compared with previous work in terms of training time and accuracy, validation, precision and recall accuracy. Table 2 shows those results for each used subset, recalling that different size-based datasets were employed to build different models in order to measure their performance.

In terms of training time, the DQ-ResNet50 model, as expected, takes more training due to its DL background and complexity. The training time takes from 5 min (Dataset 2.8k) to 1.6h (Dataset 13.4k). On the other hand, the histogram-intersection (traditional) model runs only in terms of seconds and minutes, from 12s (Dataset 2.8k) to 3.6 min (Dataset 13.4k). Unfortunately, for the model proposed in Ali *et al.* (2021) the training time is not reported.

Regarding training accuracy, the AI-based model increases its performance as the dataset is larger, from 0.780 to 0.917. Because of its model construction background, the accuracy for the traditional technique-based model is always 1. When both models are compared with Ali *et al.* (2021), the accuracy is greater, ranging from 0.991 to 0.984.

Regarding the validation stage, the performance could be measured in terms of accuracy, precision and recall. Accuracy is a general metric that measures the classifier performance for the positive and negative classes, while precision and recall are metrics that measure the accuracy of positive predictions, and the completeness of positive predictions, respectively. For more details, Table 3 shows the confusion matrices obtained for each dataset-based model, from which values for the previous metrics can be computed.

Table 2 Performance Comparison of AI and Traditional Models Based on the Training Time, Training and Validation Accuracy, Precision and Recall

Performance Metric	DQ-ResNet50	Histogram Intersection	Paper
Dataset 2.8 k			
Training Time (s)	320.536	12.297	***
Training Accuracy	0.780	1.000	0.991
Validation Accuracy	0.707	0.620	0.985
Precision	0.766	0.630	1.000
Recall	0.704	0.620	0.973
Dataset 5.6 k			
Training Time (s)	670.162	29.385	***
Training Accuracy	0.835	1.000	0.981
Validation Accuracy	0.819	0.604	0.978
Precision	0.841	0.610	0.996
Recall	0.821	0.600	0.960
Dataset 10.4 k			
Training Time (s)	3526.325	144.215	***
Training Accuracy	0.891	1.000	0.964
Validation Accuracy	0.831	0.626	0.952
Precision	0.832	0.640	0.983
Recall	0.832	0.630	0.925
Dataset 13.4 k			
Training Time (s)	4532.652	221.369	***
Training Accuracy	0.917	1.000	0.984
Validation Accuracy	0.906	0.631	0.958
Precision	0.918	0.650	0.997
Recall	0.916	0.630	0.925

Note: (***) Not reported

Source: Constructed by authors

Considering the accuracy as a more global or general metric, for the DQ-ResNest50 model the accuracy ranges from 0.707 to 0.906, as the dataset increases its size; this means that the model needs more data to generalise better and obtain a more outstanding performance. With respect to the histogram-intersection model, the accuracy does not improve as more data are added to the dataset, and the metric remains around 0.630, which is not as good a performance as DQ- ResNet50. Comparing, in terms of this metric, with the model proposed in Ali *et al.* (2021), the performance is higher, ranging from 0.985 to 0.958.

Table 3 Confusion Matrix for Each Dataset-Size-Based Model and Comparison Between AI and Traditional Techniques

		DQ-ResNet50		Histogram Intersection	
		True Class			
	C	NC		C	NC
		Dataset 2.8 k			
Predicted Class	C	226	95	181	128
	NC	69	170	99	152
		Dataset 5.6 k			
Predicted Class	C	498	94	419	302
	NC	108	419	141	258
		Dataset 8.4 k			
Predicted Class	C	756	145	663	451
	NC	181	598	177	389
		Dataset 10.4 k			
Predicted Class	C	870	176	805	543
	NC	176	858	235	497
		Dataset 10.4 k			
Predicted Class	C	1392	124	1074	723
	NC	128	1036	266	617

Source: Constructed by authors

This analysis would shows that the reference model is much better than the built ones in this work, however, before reaching such a conclusion, it is necessary to look at the models' background, specifically, the DQ-ResNet50. As stated earlier, this model is based on DQ that involves improved inference speed, lower energy consumption, easier deployment on hardware and reduced memory footprint. These characteristics should be balanced with a small decrease in model accuracy. Nevertheless, comparing both models, the accuracy is over 0.900 and the difference remains in decimal places.

A model (DQ-ResNet50) with such characteristics is much more competitive at the moment of deploying it on hardware. While it is true that the proposed model (Ali *et al.*, 2021) has higher accuracy, the complexity is higher and the deployment on hardware may cause a lot of problems.

CONCLUSIONS

In this paper, a comparison between AI and traditional-based techniques for crack detection on concrete structures has been stated. In the traditional field, a histogram-intersection model is developed, while in the AI field, a ResNet50 model based on DQ is proposed. Concerning the model's performance compared with a previous work, the proposed model in this work offers more facilities to be deployed on hardware due to DQ foundations and advantages. While it is true that the performance in the comparison model is a little higher, both models offer an accuracy of over 0.900. Additionally, the model in this work is more flexible and adaptable for hardware applications.

REFERENCES

Ai, D., Jiang, G., Lam, S.-K., He, P. and Li, C. (2023): Computer vision framework for crack detection of civil infrastructure—a review. *Engineering Applications of Artificial Intelligence*, Vol. 117, p.105478. Available at: https://doi.org/10.1016/j.engappai.2022.105478

Ali, L., Alnajjar, F., Jassmi, H.A., Gocho, M., Khan, W. and Serhani, M.A. (2021): Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors*, Vol. 21, No. 5, p.1688. Available at: https://www.mdpi.com/1424-8220/21/5/1688

Dare, P.M., Hanley, H.B., Fraser, C.S., Riedel, B. and Niemeier, W. (2002): An operational application of automatic feature extraction the measurement of cracks in concrete structures. *Photogrammetric Record*, Vol. 17, No. 99, pp.453-464.

Dorafshan, S., Thomas, R.J. and Maguire, M. (2018): SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in Brief*, Vol. 21, pp.1664-1668. Available at: https://linkinghub.elsevier.com/retrieve/pii/S2352340918314082

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2021): An image is worth 16x16 words: Transformers for image recognition at scale. In Online *International Conference on Learning Representations (ICLR)*, 3-7 May 2021. Available at: https://arxiv.org/abs/2010.11929

Dung, C.V. and Anh, L.D. (2019): Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, Vol. 99, pp.52-58.

He, K., Zhang, X., Ren, S. and Sun, J. (2015): Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp.770-778). Available at: http://arxiv.org/abs/1512.03385

Hu, G.X., Hu, B.L., Yang, Z., Huang, L. and Li, P. (2021): Pavement crack detection method based on deep learning models. *Wireless Communications and Mobile Computing*, Vol. 2021, No. 1, p.5573590.

Islam, M.M.M. and Kim, J. (2019): Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder–decoder network. *Sensors*, Vol. 19, No. 19, p.4251.

Khan, M.A.-M., Kee, S.-H., Pathan, A.S.K. and Nahid, A.A. (2023): Image Processing Techniques for Concrete Crack Detection: A Scientometrics Literature Review. *Remote Sensing*, Vol. 15, No. 9, p.2400.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. and Fieguth, P. (2015): A review on computer vision- based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, Vol. 29, No. 2, pp.196-210.

Liu, Z., Cao, Y., Wang, Y. and Wang, W. (2019): Computer vision-based concrete crack detection using u-net fully convolutional networks. *Automation in Construction*, Vol. 104, pp.129-139. Available at: https://doi.org/10.1016/j.autcon.2019.04.005

Özgenel, F. and Sorguç, A.G. (2018): Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction* (ISARC) (Vol. 35, pp.1-8). IAARC Publications.

Qu, Z., Chen, Y.X., Liu, L., Xie, Y. and Zhou, Q. (2019): The algorithm of concrete surface crack detection based on the genetic programming and percolation model. *IEEE Access*, Vol. 7, pp.57592-57603.

Singh, P., Ojha, P.N., Singh, B. and Singh, A. (2023): Crack Detection and Crack segmentation in concrete beams undergoing mode I fracture using computer vision and Convolutional Neural Network. *Canadian Journal of Civil Engineering*, Vol. 50, No. 5, pp.432-443.

Swain, M.J. and Ballard, D.H. (1991): Color indexing. *International Journal of Computer Vision*, Vol. 7, No. 1, pp.11-32. Available at: http://link.springer.com/10.1007/BF00130487

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017): Attention is all you need. In 31st Conference on *Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA. Available at: https://arxiv.org/abs/1706.03762 11pp.

Wang, A., Zhang, W. and Wei, X. (2019): A review on weed detection using ground-based machine vision and image processing techniques. *Computers and Electronics in Agriculture*, Vol. 158, pp.226-240.

Wu, H., Judd, P., Zhang, X., Isaev, M. and Micikevicius, P. (2020): *Integer quantization for deep learning inference: Principles and empirical evaluation*. Available at: https://arxiv.org/abs/2004.09602

Yamane, T. and Chun, P.-J. (2020): Crack detection from a concrete surface image based on semantic segmentation using deep learning. *Journal of Advanced Concrete Technology*, Vol. 18, No. 9, pp.493-504.

Yu, S.N., Jang, J.H. and Han, C.S. (2007): Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in Construction*, Vol. 16, No. 3, pp.255-261.

Zhang, A., Wang, K.C., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J.Q. and Li, B. (2018): Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. *Journal of Computing in Civil Engineering*, Vol. 32, No. 5, p.04018041

Zhang, A., Wang, K.C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J.Q. and Chen, C. (2017): Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 32, No. 10, pp.805-819.

BIOGRAPHY

Omer Mohamed Omer Abdalla is a Mechatronics Master's Student at the University of Oviedo, Karlsruhe University, ENSMM and Nile University under the Erasmus Mundus Programme. He holds a bachelor's degree in mechanical engineering from the University of Khartoum. His expertise spans Artificial Intelligence, Mechatronics Design and Robotics. Omer has extensive experience as a Mechanical Design Engineer. He has also served as a Research Assistant at the University of Khartoum Space Research Center and as a Teaching Assistant in Automatic Control and Robotics courses. Omer is dedicated to advancing mechanical design and mechatronics through innovative solutions and practical applications.

Ángel Encalada-Dávila is a Mechatronics Master's Student at the University of Oviedo (Spain) in the EU4M Erasmus Mundus Programme. He received his Bachelor's Degree in Mechatronics from ESPOL, Ecuador. His academic formation is related to Artificial Intelligence, Mechatronics Design, Flexible Manufacturing Systems, Biomechatronics Systems, Mobile & Articulated Robots, etc. He has wide experience in Business Analytics applied to shrimp farming and ATM hardware/software. Also, he is a researcher and works on Structural Health Monitoring and Fault Prognosis applied to Wind Turbines. His main motivation is developing innovative solutions with added value for real and complex problems.

Mitterand Sani Ekole is a Research Engineer focused on integrating deep learning, computer vision and cognitive robotics to advance healthcare. Currently, he is pursuing a Master's degree in Mechatronics, Robotics and Automation Engineering through the Erasmus Mundus Programme at Universidad de Oviedo and SUPMICROTECH- ENSMM. His roles include developing the GutSLAM algorithm at FEMTO-ST Institute and enhancing DeepStack at DeepQuest AI. He has also served as Chief Technology Officer at Unmanned Vehicles Laboratory and has experience in robotics education and drone technology. Certified in Solidworks, Python and AI foundations, Mitterand is dedicated to creating innovative AI-driven solutions for healthcare challenges.

