1. JERUSHAN J, 2. Vincent Sam JEBADURAI S, 3. Noble STAINES J, 4. SUBIN S.R 5. EBENEZER V, 6. Shamila EBENEZER A, 7. HEMALATHA G

Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India
ORCID: 1. 0009-0005-8596-2560; 2. 0000-0002-9820-8103; 3. 0009-0003-4372-3620;
5. 0000-0002-0801-6926; 6. 0000-0002-5167-8968; 7. 0000-0001-7067-3786.

doi:10.15199/48.2025.02.32

Crack characterisation in buildings utilizing deep learning techniques

***Abstract**. A significant field of research is the use of deep learning algorithms to detect and characterize cracks in structures. Building cracks may cause catastrophic structural collapses that endanger people's lives and property. This issue can be helped by deep learning algorithms, which allow for the very accurate identification and categorization of various crack forms. The present study uses a data set of 5000 photos to examine how image pre-processing affects the effectiveness of Deep Learning crack detection. The outcomes demonstrated that the CNN model's ability to identify cracks in concrete buildings is unaffected by the use of a pretrained model with RGB weights. Pretrained VGG16 and the Keras Python library are used to create a CNN model. The SciKit Image Python package was employed to divide the original picture data set into five comparison sets. The created model performed better than 98% in terms of accuracy and F1 values.*

***Streszczenie.** Istotnym obszarem badań jest wykorzystanie algorytmów głębokiego uczenia do wykrywania i charakteryzowania pęknięć w konstrukcjach. Pęknięcia w budynkach mogą powodować katastrofalne zawalenia konstrukcyjne, które zagrażają życiu i mieniu ludzi. Problem ten można rozwiązać za pomocą algorytmów głębokiego uczenia, które umożliwiają bardzo dokładną identyfikację i kategoryzację różnych form pęknięć. W tym badaniu wykorzystano zestaw danych 5000 zdjęć, aby zbadać, w jaki sposób wstępne przetwarzanie obrazu wpływa na skuteczność wykrywania pęknięć metodą głębokiego uczenia. Wyniki wykazały, że zdolność modelu CNN do identyfikowania pęknięć w betonowych budynkach nie jest naruszona przez użycie wstępnie wytrenowanego modelu z wagami RGB. Wstępnie wytrenowany VGG16 i biblioteka Keras Python są używane do tworzenia modelu CNN. Pakiet SciKit Image Python został użyty do podzielenia oryginalnego zestawu danych obrazu na pięć zestawów porównawczych. Utworzony model uzyskał wyniki lepsze niż 98% pod względem dokładności i wartości F1. (**Charakterystyka pęknięć w budynkach z wykorzystaniem systemów głębokiego uczenia**)*

Keywords: Characterisation, cracks, deep learning, CNN, python
Słowa kluczowe: Charakterystyka, pęknięcia, głębokie uczenie, CNN, python

1. Introduction

Deep learning models demonstrate competence in handling large-scale datasets in order to identify and describe structural flaws in structures. This is mostly achieved via sophisticated image analysis techniques, in which deep learning models examine building photos to identify cracks of different shapes, sizes, and sorts. The crack detection strategies or techniques involve segmentation, classification, and detection procedures. Segmentation is the computational process of identifying and separating the different components within a picture. In building inspection, the segregation of the structurally damaged, like the cracked sections from the intact parts, is called segmentation. Classification is the process of systematically assigning descriptive categories to areas that have been separated out of a picture. In this application, it refers to different categories of cracks based on their unique features. Detection refers to the act of determining the exact position and dimensions of cracks found in the building structure.

It will be powered by some of the deep learning techniques that identify and characterize key structural faults in building infrastructural facilities, including CNNs, RNNs, and DBNs. These algorithms perform very well at precisely identifying and classifying the various types of cracks that appear in structures using large datasets. Deep learning algorithms have completely changed the procedures for building crack detection, classification, and characterization. The mentioned techniques can project preventive measures to prevent structural collapses and reduce hazards to health and property integrity by correctly assessing and examining the problem structures.

The following research makes use of deep learning algorithms in the processes of identification, localization, and rating of the severity of structure cracks in buildings. Principally, this aims at improving the present structural safety measures against a number of dangers and reducing the risks of accidents caused by the loss of integrity of buildings.

1.1 Insights from Literature

This has formed the basis of numerous studies regarding the use of deep learning techniques in building crack detection.

For the detection of cracks in concrete image, [1] proposed a deep learning approach using a convolutional neural network and a deep belief network for crack classification and detection purposes. [2], [3] developed a deep learning framework for identifying cracks from pavement photos. Multiple deep neural networks recognize and classify different kinds of pavement cracks. By the same token, [4] proposed a deep learning approach that identifies and classifies surface cracks in concrete buildings with a neural network architecture together with a backpropagation algorithm. On their part, [5] proposed an inventive deep learning system for pavement fracture detection. Their method used a CNN, which carefully scanned image patches to ensure an accurate classification of crack and non-crack areas. [6] put forward the "VGG16" method, a deep learning-based crack detection and classification methodology for identifying and classifying different fracture types. [7] provided a deep learning approach for the identification of cracks in pavement images. Their approach could correctly classify fracture and non-fracture images using a design for a convolutional neural network. In [8], integrated multi-feature learning with multi-scale fusion approaches in a deep learning framework for the identification of concrete cracks. Their approach performed better compared to traditional machine learning techniques. Similarly, [9] proposed the framework of "AC-CNN," which has been a breakthrough of deep learning techniques designed for the identification of cracks in building façade photographs. It demonstrates potent ability in the detection and classification of different fracture types.

Besides, [10] conducted a study on detecting cracks in concrete structures using convolutional neural networks and transfer learning. Their study was focused on the dataset composed of images from concrete specimens showing three different classes of fracture severity: no cracks, fine cracks, and severe cracks. The outcome shows how good the CNN model performed in the detection of both small and large cracks in the concrete sample, with an accuracy value that is high. In the experiment, [11] applied deep learning techniques CNNs and RNNs for identifying road cracks. With the aid of an image-based dataset dedicated to road pavement cracks, the research demonstrated that deep learning algorithms could be very effective in the identification and classification of those structural flaws. On the other hand, [12] suggested a deep learning model of CNNs for the purpose of detecting cracks in reinforced concrete buildings. To be more specific, the algorithm was trained on a large collection of photos showing different kinds and sizes of cracks in reinforced concrete. The results showed how fine the algorithm could identify and classify the cracks in these structural components. [13] proposed a deep learning framework that ensembles CNNs in order to automatically identify cracks on building facades. This framework showed leading performance in the accurate identification and classification of different kinds of cracks. [14] have worked on the application of transfer learning in training deep neural networks toward the identification of cracks in concrete buildings. The results showed how their approach could identify and classify a large number of crack types in the concrete pavement. [15] have just come up with a deep learning approach for automated detection and separation of cracks in concrete buildings. In the identification and segmentation tasks, this achieved more than 97% accuracy with the aid of a dataset containing over 2,500 photos of concrete surfaces bearing different kinds of cracks.

It is observed from the literature that deep learning approaches using CNNs successfully identify and characterize the structural cracks in buildings and other infrastructures. The precision of the proposed models depends on the amount and quality of data used for training purposes. Accordingly, further research into building context fracture detection models is expected to be more accurate and effective. This research effort is intended to approach the systematic study of photographic evidence of cracks on different surfaces using deep learning techniques.

2. Cracks in Construction

2.1 Cracks and its classification

Cracks in concrete and masonry are characterized by the complete or partial separation of the material into two or more pieces due to cracking or spalling. These cracks can occur in either plastic or hardened concrete. Cracks can have a wide range of reasons, from minor cosmetic problems to major structural flaws and durability issues. Therefore, cracks can serve as indicators of the overall extent of visible damage or hint at more profound structural problems. Cracks are categorized into five main categories in a systematic manner: structural cracks, which are crucial for maintaining the structural integrity of the building; concrete cracks, which are specific to the concrete material; plaster cracks, which happen on plastered surfaces; joint cracks, which show up at the intersections of various construction elements or materials; and floor cracks, which are specific to flooring systems. Understanding the type and cause of each crack is essential for diagnosing the underlying issues and implementing appropriate remedial measures.

2.1.1 Structural cracks

Cracks exceeding 1/8 inch (3 mm) in width are classified as structural cracks (Fig.1), typically manifesting as horizontal, diagonal, or stepped formations with nearly symmetrical patterns. Primary causes include water-saturated ground following heavy rainfall, inadequately prepared construction sites, planning errors, ground movement, and soil shrinkage due to prolonged drought conditions. The emergence of these structural cracks can lead to functional issues such as difficulty in closing windows and doors, and noticeable tilting of floors, significantly diminishing the comfort and usability of a home. Moreover, the discussed cracks compromise the structural integrity of the building, adversely affect indoor air quality, create conditions conducive to mould growth, and provide entry points for vermin and crawling insects. Promptly addressing structural cracks is essential to maintain the stability, safety, and habitability of the structure.

Fig.1. Structural Crack

2.1.2 Concrete Crack

Outdoor concrete structures often undergo shrinkage during the curing process, primarily due to the evaporation of water from the concrete mixture. Cracking (Fig.2) occurs when the force of this shrinkage exceeds the concrete's inherent strength. This phenomenon can be conceptualized as a race between two competing processes: the evaporation of water and the gradual strengthening of the concrete over time.

Fig. 2. Concrete Crack

2.1.3 Joint crack

A construction joint (Fig.3) is strategically incorporated into a structure to minimize potential cracking and to streamline the construction process. It serves as an engineered point of controlled movement and is essential in accommodating the natural expansion and contraction of materials. Without construction joints, structural integrity could be compromised, leading to undesirable cracks and deformations. Well-designed construction joints enable efficient construction without the need for costly and time-consuming remedial procedures and techniques.

Fig.3. Joint Crack

2.1.4 Plaster crack

Plaster cracks, depicted in Fig.4, manifest as minor cracks in plaster walls. Despite popular belief, these kinds of cracks are very common and usually harmless. These cracks generally appear when gypsum plaster is drying and curing; a small amount of shrinkage causes this to happen. These cracks are frequently seen in newly built or recently enlarged structures as a result of continuous settling processes, which can take one to three years to fully settle. Plaster surface cracks can also arise as a result of variables including moisture content, temperature fluctuations, and ambient humidity. The mentioned climatic variations make the plaster expand and contract, enabling movement away from or toward the general structural stiffness of the building. These small gaps may later develop into larger, more visible cracks as the plaster settles and cures.

Fig. 4. Plaster Crack

2.1.5 Floor crack

Most often, floor cracks (Fig.5) are an indication of poor building techniques or a substrate/uneven foundation that has slipped under the pressure exerted by the structure of the building. Such issues may escalate to bring many problems to the stability and integrity of a structure over some time. Specifically, there is a big risk in settling foundations that manifest in the expensive gradual degradation of floor surfaces. Such degeneration can be seen with cracks on the walls, misaligned walls, or drooping floors. What might look like tiny cracks may later turn into a full-fledged structural issue. Sagging buildings, sticking doors and windows, and standing water next to cracks indicate more serious structural instability. The concerns must hence be looked into with immediate effect, as they may escalate and worsen inherent structural defects if not addressed promptly, resulting in time-consuming and costly corrective action.

Fig.5. Floor Crack

3. Deep Learning algorithm

3.1 Deep Learning

A class of machine learning algorithms, "deep learning" is inspired by how the human brain is constituted and its operations, but with very differentiated capabilities. Huge amounts of data can be processed and used to train these algorithms. While a neural network with just one layer can make some basic predictions, several hidden layers bring accuracy and deeper tuning. Deep learning has a variety of applications that realize many artificial intelligence services and systems, thereby enabling automation and running complex analyses autonomously. Nowadays, these technologies underlie voice-activated devices, like TV remotes and digital assistants, and complex applications for credit card fraud detection. Moreover, deep learning is fuelling innovation in very cutting-edge areas, like autonomous vehicles.

3.2 Mechanisms of Deep Learning

Artificial neural networks, also known as deep learning neural networks, are a means of analyzing data inputs through the use of connected nodes that are biased and weighted. This makes them capable of emulating some features of the way a brain works. Within this context, it becomes possible to recognize, identify, and describe patterns present within datasets with accurate precision.

A deep neural network has a large number of layers, and each of them improves the results on classification or prediction using the one preceding it. This computation-based development of a network is called forward propagation. The input and output layers are normally the visible levels to a deep neural network. The input layer collects and processes data; the output layer then makes the final predictions or classifications.

Backpropagation is a method that uses algorithms such as gradient descent to compute the prediction error to optimize the performance of the network. Computed information gets propagated backward through layers to update the weights and biases. With time, predictions get optimized, and mistakes are minimized when forward and backpropagation is used repeatedly. These techniques are manifold and in turn complicated, even if the above is a simple deep neural network.

There exist many types of neural networks. Some of them specialize in diverse areas of data and problem domains. Convolution Neural Networks, for example, have been surpassing humans since 2015 in specific computer vision tasks by detecting properties in images for tasks such as recognizing objects. On the other hand, recurrent neural networks are very good in processing sequential or time series data, so they have applications in speech recognition and natural language processing.

3.3 Artificial Neural Network

Neural networks are designed to resemble the way the human brain processes information. Artificial Neural Networks (ANNs) are usually composed of many artificial neurons arranged in linked layers. Every neuron in an Artificial Neural Network (ANN) functions as a mathematical function, taking in input values, processing them using an activation function, and then sending the results to other neurons or to the ultimate output. In essence, a neural network consists of activation functions that decide the output depending on the inputs received, output layers for generating results, and input layers for consuming data. A graphical depiction of this mathematical model is shown in Fig.6.

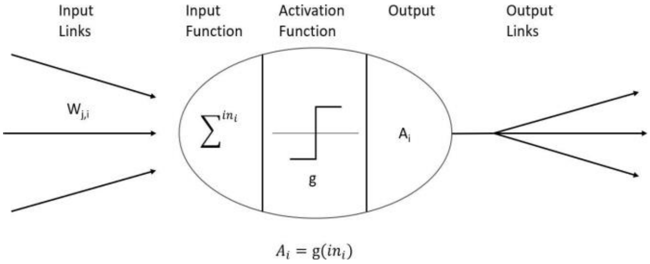

Fig.6. Mathematical model of artificial neuron

3.4 Convolutional Neural Network

A subclass of deep neural networks called convolutional neural networks (CNNs) is mostly used for processing visual data using computational fields of view. CNNs are typically designed with convolutional layers, subsampling layers, and a densely linked layer that generates the final softmax vector. This architectural paradigm has its roots in LeNet-5, an innovative ConvNet designed for 28x28 grid handwritten digit recognition. Several blocks that individually mimic the traditional LeNet-5 design are common in modern CNN architectures, allowing for the creation of models with hundreds of layers of depth.

3.5 Texture classification

Texture classification is one of the most difficult problems in machine learning since it differs from object-based classification techniques. Statistical characteristics and recurring patterns, varying from extremely regular to stochastic, define textures. Classification, segmentation, synthesis, and form analysis are the four main sub-problems that comprise the area of texture analysis; this project focuses mostly on classification tasks.

In the 1980s and 1990s, filtering techniques and statistical modelling were the two key areas of focus for early texture analysis research efforts. The main goal of filtering algorithms is to extract textural information through the use of simpler filters like Gaussian differencing or convolutional filters (like Gabor filters and pyramidal wavelets). Statistical modelling, on the other hand, uses probability distributions seen in random fields to describe textures.

Convolutional neural networks (CNNs), attribute-based classification systems, and Bag-of-Words (BoW) models are the three main approaches that have emerged over time for texture classification. An important paper, by [16] highlighted CNNs' enduring bias for challenges including texture. The article also included research showing that, on datasets such as ImageNet, some CNN architectures may perform robust object classification with just texture information. For example, [17] found that linear classifiers run on CNN texture maps showed less loss divergence than the original network, therefore indicating that CNNs are good at capturing and utilizing texture information in a very natural way. [16] elaborated that CNNs might actually integrate texture signals across a number of layers, even with tiny receptive fields, and actually perform well in highly accurate ways on ImageNet tasks relating to object categorization. This reflects the capability of CNNs in conditions where texture-based signals are key, hence making them very strong contenders for the more complex classification problems like crack detection.

4. Model Development using Software Tools

4.1 Software Tools Used in the Experiment

Several software tools were used throughout the experimental phase for easier collection of data and testing on the models. This section explains the various software tools that have been in use within the experimental processes of this study, outlining what they contribute to the research aims and how they enable a reliable method for data collection and experimentation.

4.1.1 Tensor flow

TensorFlow is a very popular platform for machine learning, with a large and extensive ecosystem that includes standardized resources, libraries, and tools. The platform reduces the need to study deeper knowledge of complicated characteristics by providing ease of building machine learning algorithms with accessibility. In this respect, one can easily experiment with different designs for a model and parameters within the TensorFlow platform. It also provides the ability to train models more efficiently due to its usage of GPU acceleration, speeding up training times.

4.1.2 Keras

Keras is an extremely high-level neural network API that can run on top of TensorFlow, CNTK, or Theano. Keras' aim is to make it easy to experiment in machine learning. It encourages speed from ideation to execution in order to facilitate fast iteration and faster creation of results. Sequentially created Keras models are characterized by their very self-explanatory coding paradigm; that is, it is very easy to implement new functions, modules, loss functions, or activation functions. Well, Keras is a Python implementation that guarantees readable and debug-friendly code. This makes it possible to have fast development and debugging processes. The Keras API offers support for using pre-trained models learned on ImageNet. Therefore, this creates flexibility in deploying models either with or without a classification layer, depending on specific research or application requirements.

4.2 Model Development of Crack Detection

4.2.1 Data Collection

The performance of machine learning models is strongly influenced by the quantity and makeup of a dataset. Although smaller photos speed up training and minimize data quantity, too tiny images could not contain enough information to support a thorough analysis. For example, the VGG16 architecture can handle somewhat bigger dimensions but usually uses a default input size of 224x224 pixels. This study used a dataset that was designed to categorize photos of building cracks. There are 5000 photos in this collection, and each one has 227x227 pixels in RGB channels. Pictures are divided into two classes: positive (crack) and negative (crack). There are 1000 photos in each category. The basic resource for training and testing algorithms targeted at fracture detection in building structures is this organized dataset.

4.2.2 Image Processing

After being imported, each image was processed separately with TensorFlow to create structured components. Then, with specific code, the Sci-Kit Image module in Python 3.8 was used to further improve and preprocess the photos. To be more precise, methods for improving crack features were used in the first stages of image processing (IP) crack detection. Deep learning (DL) approaches typically forgo conventional picture pre-processing methods in order to reduce the amount of data needed for algorithmic learning. On the other hand, image processing techniques were used in this work to create four unique datasets that were designed for further analysis and model training. These datasets are very important in checking how well the deep learning models identify cracks in building photos, varying by light and scenario circumstances.

4.2.3 Proposed CNN Model

In this research, a pre-trained CNN architecture was used. On the other hand, transfer learning using pre-trained architectures has been shown to significantly enhance model performance for varied fields. Because of its very good performance on ImageNet benchmarks, VGG16 just happens to be one of the pre-trained architectures mostly used for crack detection tasks. This paper proposes the CNN-based architecture, where fully connected layers are sprinkled over a series of convolutional blocks. Each block also contains a convolutional layer for filtering the output from the previous layer, with an activation unit supplying non-linearity and a pooling layer down-sampling the feature maps. The convolutional layers extract spatial information relevant to fracture detection by convolving the kernels or filters over the input data. This will ensure that the model efficiently acquires discriminative features from input photos and improves its capacity of detecting cracks in building structures with accuracy.

4.2.4 Model Development

All computational activities were done using a laptop equipped with RAM of 16GB and an AMD Ryzen 7 4800H CPU with Radeon Graphics, running at 2.90GHz. It had an NVIDIA GeForce GTX 1650 Ti GPU running a 64-bit operating system for creating the models. Anaconda Spyder 4.1.5 was used as the development environment with Python 3.8. Python libraries used in this work were Keras 2.4 and Tensorflow 2.5. With this, convolutional neural networks could be developed with Python scripts or Jupyter notebooks for a couple of environments. Almost all deep learning models were available with the Keras library, like the VGG16 model by [18] Specifically, this network was employed for the purpose of this research work in crack detection applications. Max pooling was done in order to enhance the performance in crack identification. The model will be trained with the Adam optimizer during a pre-defined training period. It uses a binary cross-entropy loss function, and training will be done on the entire dataset by setting 200 photos per batch for each epoch. Fifty epochs of training are used so that the model learns properly and converges. The model's input layer specified variables like the size of the image, the number of channels, and the details of the dataset. To increase the amount of the dataset and reduce orientation biases, data augmentation techniques included flipping the data vertically and horizontally, rotations (0.2/2π), and sequential shifts.

In order to make the pixel values (which range from 0 to 255) compatible with the ImageNet weights used for pre-training, a normalizing layer was also included. The pooling layers of the VGG16 architecture were used to transform the outputs of the basic model into vector representations once it was initialized with ImageNet weights. In order to improve model generalization, dropout regularization was used, which involves randomly deactivating filters during training. The model outputs were combined in the last dense classification layer to provide binary predictions, which assigned a value of 1 to cracked areas and a value of 0 to non-cracked areas. Following prediction, a rounding function transformed discrete crack predictions 1 from probabilistic outputs 0.5. Taking everything into account, these configurations and techniques ensured that the VGG16 model was used consistently for effective crack detection.

4.2.5 Model Analysis

In this model study, the performance was evaluated based on how it had classified the data on binary class labels represented by the letters positive and negative. There were four other resulting outputs for each input case whereby the classifier predicted classes as either Yes or No. The false-positive predictions were represented by the false-positive instances, and on the other hand, the false-negative predictions were represented by the false-negative instances. True positives represented the correct prediction of positive cases, while true negatives indicated correct forecasts of negative cases. For evaluation, in this paper, the following are used: Accuracy, True Negative Rate, Positive Predictive Value, Negative Predictive Value, True Positive Rate, and the F1 score, which is the harmonic average of recall and precision. These metrics give an all-rounded description of the performance of the CNN classifier. The metrics have been computed in conventional ways based on TP, TN, FP, and FN. The comparison is also made with a baseline control to determine any improvements brought about by the CNN-based method, so as to validate its efficacy in enhancing classification accuracy and reliability.

5. Results and Discussion

By leveraging transfer learning in the development of multiple instances of CNN models, five different CNN models were obtained from two trainings using different image datasets. Each model was used to output a confusion matrix for comparison and evaluation of their performance. This section presents the performance measures produced based on the analysis of the confusion matrix run against the test dataset, which contains 5000 sample pictures. The models were trained with validation data, and training accuracy was plotted over several epochs to show how the models developed and improved. For testing, RGB data was used as the control group. Somewhat surprisingly, the pre-processed photos contained only one brightness channel, and yet the pretrained VGG16 model used RGB weights pre-trained on three channels: red, green, and blue. This was in anticipation of enhancing model performance through the expected effect of colour dependence on RGB values.

5.1 Training of 50 EPOCHS

To check for the probable benefits of longer training with respect to model accuracy in processing single-channel images, a training regimen of 50 epochs was used. Now, analyzing the confusion matrix, it is without exception that all such models that underwent such a prolonged train period outperformed their sibling models trained for fewer epochs. This discovery highlights how extensive training sessions are effective in improving the model's capacity to correctly analyze and categorize single-channel picture input.

Fig.7 presents the graphical representation of the accuracy achieved over 50 epochs. Fig.8 displays the confusion matrix obtained after 50 epochs of training. Fig.9 showcases the accuracy per class achieved after 50 epochs of training.

These figures collectively provide a comprehensive visual analysis of the model's performance metrics throughout the training process.

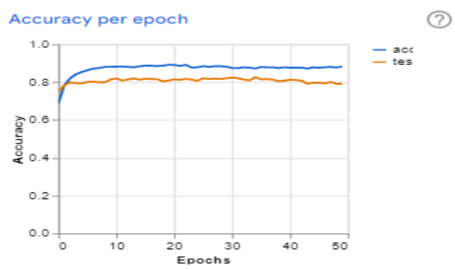

Fig.7. 50-Epochs Accuracy

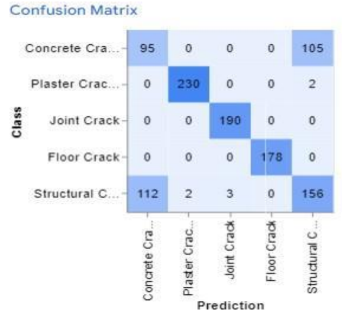

Fig.8. 50-Epochs Confusion Matrix

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Concrete Crack	0.47	200
Plaster Crack	0.99	232
Joint Crack	1.00	190
Floor Crack	1.00	178
Structural Crack	0.57	273

Fig.9. 50-Epochs Accuracy Per Class

5.2 Structural Crack Detection

It has been trained on a huge number of sample photos to aid in crack detection. Fig.10 shows how, when fed into the program, a structural crack image is turned into a percentage-based bar graph. This graphical presentation clearly indicates the classification result, which marks the type of crack recognized as structural crack and as Concrete crack with 77% and 33% confidence level respectively.

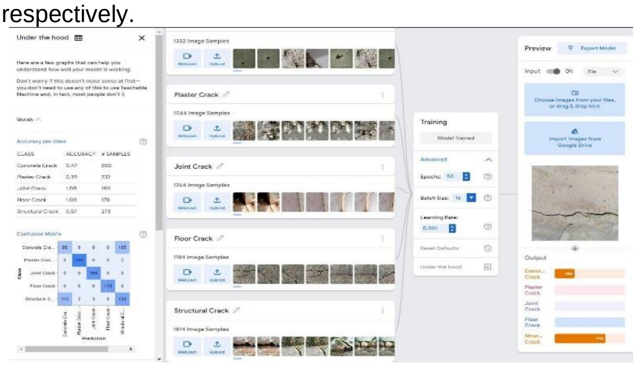

Fig.10. Structural crack detection

5.3 Concrete Crack Detection

Fig.11 illustrates the process of uploading a concrete crack image into the program with a large sample size and extensive training so that a bar graph may be plotted on a percentage basis for it to perform crack detection. Through this graphical representation, the result of the classification is successfully shown, such as the type of crack being identified as a concrete crack with a confidence level of 100%.

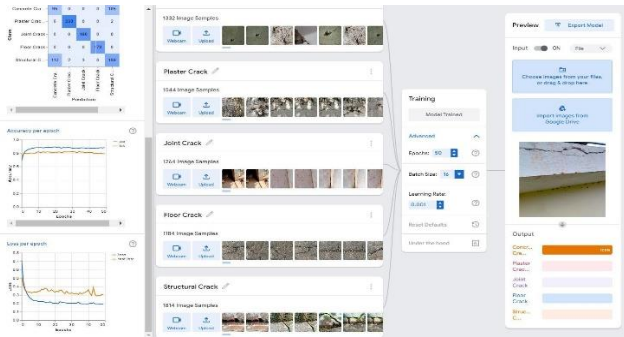

Fig.11. Concrete crack detection

5.4 Floor Crack Detection

Fig.12 illustrates the processing of a floor crack image that has been uploaded into the program, producing a bar graph with percentage basis using a large sample size of photographs. The image dataset has undergone extensive training to enable the crack identification. This graphical depiction, showing the type of crack that was determined as being a floor crack with 100 percent certainty level, effectively demonstrates the categorization findings.

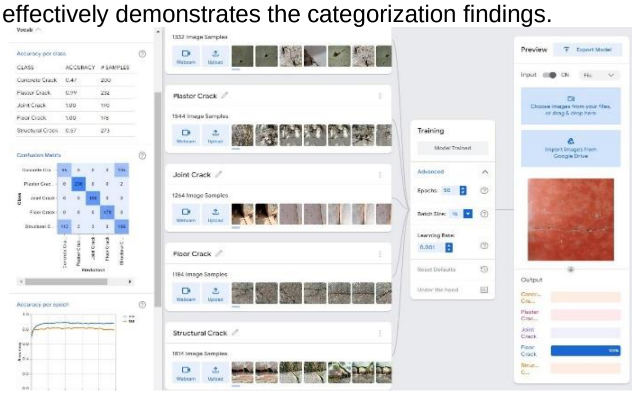

Fig.12. Floor crack detection

5.5 Joint Crack Detection

Fig.13 interprets step-by-step procedure to process a joint crack image uploaded into the tool in order to produce a bar graph with a percentage basis using a large sample number of photos. To facilitate the detection of cracks, the picture dataset underwent intensive training. This graph imitates the classification findings very well, stating that it is a joint crack with 100% certainty.

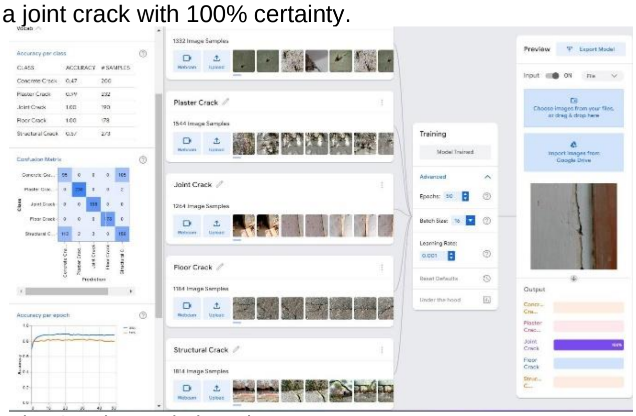

Fig.13. Joint crack detection

5.6 Plaster Crack Detection

Fig.14 shows how a bar graph with a percentage basis can be derived from a plaster crack image that was inputted into the program using a large sample number of photographs. The dataset image was then trained exhaustively to allow the system to identify the cracks. This graphical display served well in presenting the classification results by indication of the type of crack that was identified and classified to be a plaster crack with 100% certainty.

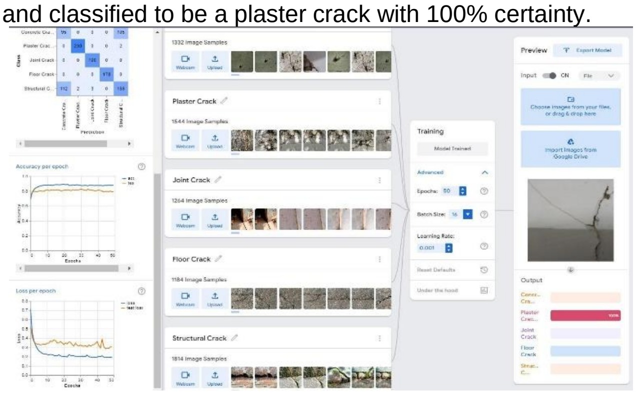

Fig.14. Plaster crack detection

6. Conclusion

This study examined the effects of image pre-processing methods on deep learning (DL) crack detection ability using a dataset of 5000 images. The results showed that the CNN model's capacity to identify cracks in concrete structures was not significantly impacted by the use of a pretrained model with RGB weights. A CNN architecture was built using the SciKit Image Python library, and the original photo dataset was split up into five different comparison sets using the Keras Python package and pretrained VGG16. The model's F1 score after 50 epochs was 99.549%, which is similar to the RGB model's score of 99.533% and highlights the color independence of crack detecting features. Although color had little influence, the study showed possible biases induced by RGB-weighted pretrained models, hence future work should validate using fully segmented and trained models. The study's analysis was carried out with pretrained models using RGB weights, concentrated on the three RGB channels. Subsequent studies are to examine the procurement of weights particular to each channel and assess the influence of IP-FCN pixel segmentation on pixel precision. Furthermore, contrasting different pretrained models like VGG11, VGG19, and AlexNet may provide information on how effectively they perform in tasks involving deep learning based crack detection.

Acknowledgement: The resources necessary for this investigation were made available to the authors by the Karunya Institute of Technology and Sciences.

***Authors**: Mr. Jerushan J, Research Scholar, Department of Civil Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: jerushanj21@karunya.edu.in*

Dr. Vincent Sam Jebadurai S, Assistant Professor, Department of Civil Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: vincent@karunya.edu

Mr. Noble Staines J, Post Graduate Student, Department of Civil Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: noblestaines21@karunya.edu.in

Mr. Subin S R, Under Graduate Student, Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114.

Dr. Ebenezer V, Assistant Professor, Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: ebenezerv@karunya.edu

Dr. Shamila Ebenezer A, Assistant Professor, Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: shamila_cse@karunya.edu

Dr. Hemalatha G, Former Professor, Department of Civil Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Email: hemagladston@gmail.com

REFERENCES

[1] R. Fan et al., “Road crack detection using deep convolutional neural network and adaptive thresholding,” IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2019-June, pp. 474–479, Jun. 2019, doi: 10.1109/IVS.2019.8814000.

[2] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” Proceedings - International Conference on Image Processing, ICIP, vol. 2016-August, pp. 3708–3712, Aug. 2016, doi: 10.1109/ICIP.2016.7533052.

[3] W. Ouyang and B. Xu, “Pavement cracking measurements using 3D laser-scan images,” Meas Sci Technol, vol. 24, no. 10, p. 105204, Aug. 2013, doi: 10.1088/0957-0233/24/10/105204.

[4] D. Joshi, T. P. Singh, and G. Sharma, “Automatic surface crack detection using segmentation-based deep-learning approach,” Eng Fract Mech, vol. 268, p. 108467, Jun. 2022, doi: 10.1016/J.ENGFRACMECH.2022.108467.

[5] H. N. Nguyen, T. Y. Kam, and P. Y. Cheng, “An Automatic Approach for Accurate Edge Detection of Concrete Crack Utilizing 2D Geometric Features of Crack,” J Signal Process Syst, vol. 77, no. 3, pp. 221–240, Oct. 2014, doi: 10.1007/S11265-013-0813-8/METRICS.

[6] R. Ali, J. H. Chuah, M. S. A. Talip, N. Mokhtar, and M. A. Shoaib, “Structural crack detection using deep convolutional neural networks,” Autom Constr, vol. 133, p. 103989, Jan. 2022, doi: 10.1016/J.AUTCON.2021.103989.

[7] G. X. Hu, B. L. Hu, Z. Yang, L. Huang, and P. Li, “Pavement Crack Detection Method Based on Deep Learning Models,” Wirel Commun Mob Comput, vol. 2021, no. 1, p. 5573590, Jan. 2021, doi: 10.1155/2021/5573590.

[8] J. Zhong, J. Zhu, J. Huyan, T. Ma, and W. Zhang, “Multi-scale feature fusion network for pixel-level pavement distress detection,” Autom Constr, vol. 141, p. 104436, Sep. 2022, doi: 10.1016/J.AUTCON.2022.104436.

[9] P. Wang et al., “An automatic building façade deterioration detection system using infrared-visible image fusion and deep learning,” Journal of Building Engineering, vol. 95, p. 110122, Oct. 2024, doi: 10.1016/J.JOBE.2024.110122.

[10] V. P. Golding, Z. Gharineiat, H. S. Munawar, and F. Ullah, “Crack Detection in Concrete Structures Using Deep Learning,” Sustainability, vol. 14, no. 13, p. 8117, Jul. 2022, doi: 10.3390/su14138117.

[11] V. Mandal, L. Uong, and Y. Adu-Gyamfi, “Automated Road Crack Detection Using Deep Convolutional Neural Networks,” in 2018 IEEE International Conference on Big Data (Big Data), IEEE, Dec. 2018, pp. 5212–5215. doi: 10.1109/BigData.2018.8622327.

[12] K. C. Laxman, N. Tabassum, L. Ai, C. Cole, and P. Ziehl, “Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning,” Constr Build Mater, vol. 370, p. 130709, Mar. 2023, doi: 10.1016/J.CONBUILDMAT.2023.130709.

[13] K. Chen, G. Reichard, X. Xu, and A. Akanmu, “Automated crack segmentation in close-range building façade inspection images using deep learning techniques,” Journal of Building Engineering, vol. 43, p. 102913, Nov. 2021, doi: 10.1016/J.JOBE.2021.102913.

[14] R. E. Philip, A. D. Andrushia, A. Nammalvar, B. G. A. Gurupatham, and K. Roy, “A Comparative Study on Crack Detection in Concrete Walls Using Transfer Learning Techniques,” Journal of Composites Science 2023, Vol. 7, Page 169, vol. 7, no. 4, p. 169, Apr. 2023, doi: 10.3390/JCS7040169.

[15] J. J. Kim, A.-R. Kim, and S.-W. Lee, “Artificial Neural Network-Based Automated Crack Detection and Analysis for the Inspection of Concrete Structures,” Applied Sciences, vol. 10, no. 22, p. 8105, Nov. 2020, doi: 10.3390/app10228105.

[16] R. Geirhos, C. Michaelis, F. A. Wichmann, P. Rubisch, M. Bethge, and W. Brendel, “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness,” 7th International Conference on Learning Representations, ICLR 2019, Nov. 2018, Accessed: Jul. 21, 2024. [Online]. Available: https://arxiv.org/abs/1811.12231v3

[17] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 2414–2423, Dec. 2016, doi: 10.1109/CVPR.2016.265.

[18] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, Sep. 2014, Accessed: Jul. 21, 2024. [Online]. Available: https://arxiv.org/abs/1409.1556v6