Characterization of Cracks in the Building Using Deep Learning Technique

Noble Staines J[1], Justin Moses Selvamony[2]

[1] *PG Student, Department of Civil Engineering, Karunya Institute of Technology and Science*
[2] *Professor, Department of Civil Engineering, Toc H Institute of Science and Technology*

---***---

Abstract - Detecting and characterizing cracks in buildings using deep learning techniques is a crucial research area. Cracks in buildings could lead to catastrophic structural failures, which can be hazardous to human life and property. Deep learning techniques can aid in addressing this problem by enabling the detection and classification of different types of cracks with high accuracy. This study investigates the effect of image pre-processing on the performance of DL crack detection using a data set of 5000 images. The results showed that using a pretrained model with RGB weights does not affect the performance of the CNN model for detecting cracks in concrete structures. A CNN model is developed using the Keras Python package and pretrained VGG16. The original image data set was converted into five comparison sets using the SciKit Image Python package. The performance of the developed model in terms of accuracy and F1 values were all above 98%.

***Key Words*:** Characterisation, Cracks, Deep learning, CNN, Python, SciKit.

1. INTRODUCTION

Deep learning models are capable of analysing large amounts of data to detect and identify structural defects in buildings. This can be achieved through image analysis techniques, where deep learning models analyze images of buildings to identify cracks of different types, shapes, and sizes. The techniques used to identify cracks in buildings include segmentation, classification, and detection. Segmentation is the process of identifying and separating different parts of an image. In this case, it involves separating the cracked parts of a building from the unaffected areas. Classification is the process of assigning a label to the segmented parts of the image. In this case, the labels could be the different types of cracks. Detection is the process of identifying the location and size of the crack in the building.

Deep learning algorithms such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) are widely used in identifying and characterizing cracks in building structures. These algorithms are capable of learning from large amounts of data, enabling them to detect and classify different types of cracks in buildings accurately. Deep learning techniques have the potential to revolutionize the way we identify, classify and characterize cracks in buildings. By enabling us to detect and analyze the state of building structures accurately, we can take preventive measures to avoid structural failures and reduce the risks to human life and property.

This research involves detecting the presence, location, and severity of cracks in a building using deep learning algorithms. The primary objective of this research is to help prevent structural failures and avoid potential accidents that may result from cracked buildings.

Numerous studies have been conducted to explore the use of deep learning techniques for the detection of cracks in buildings. One of the study by Kim et al. (2011) which proposed a deep learning-based approach for the detection of cracks in concrete images. The proposed method used a deep belief network (DBN) and a convolutional neural network (CNN) for the detection and classification of cracks. Another study by Ouyang et al. (2011, 2013)) presented a deep learning-based framework for crack detection in pavement images. The proposed method used multiple deep neural networks to detect and classify different types of pavement cracks. Similarly, in a study by Kim et al. (2013), a deep learning-based method was proposed for the detection of surface cracks in concrete structures. The proposed approach used a neural network architecture and a backpropagation algorithm for the detection and classification of cracks. Nguyen et al. (2014), presented a novel deep learning-based framework was proposed for the detection of pavement cracks. The proposed method used a convolutional neural network (CNN) to analyze the image patches and classify them into crack or non-crack regions with high accuracy. Bi et al. (2017) proposed a deep learning-based method called "VGG16" to detect and classify different types of cracks. In addition, there was a study by Wang et al. (2017) which proposed a deep learning-based approach for the detection of cracks in pavement images. The proposed framework used convolutional neural network architecture to classify images with or without cracks. In a study by Zhang et al. (2018), a deep learning-based framework was proposed for the detection of concrete cracks, which incorporated multi-feature learning and multi-scale fusion. The proposed method achieved better performance compared to traditional machine learning methods. Similarly, in a study by Liu et al. (2018), a new deep learning-based framework called "AC-CNN" was proposed for the detection of cracks in images of building facades. The proposed framework was able to effectively detect and classify various types of cracks with high accuracy.

One such study was conducted by Salleh et al. (2020), who used a combination of convolutional neural networks (CNNs-) and transfer learning to detect cracks in concrete structures. The study involved the use of a dataset of images of concrete specimens, with each specimen having either no cracks, fine cracks, or severe cracks. The results showed that the CNN model was able to achieve a high level of accuracy in detecting fine and severe cracks in the concrete specimens.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Special Edition - Volume 09 ICITES -2025 SJIF Rating: 8.586 ISSN: 2582-3930

In another study, Huang et al. (2019) employed deep learning models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for the detection of road cracks. The study used an image-based dataset of road pavement cracks and showed that deep learning algorithms could accurately detect and classify road pavement cracks. In a study by Ouali et al. (2020), a deep learning model was used to detect cracks in reinforced concrete structures. The model was based on a convolutional neural network (CNN) and was trained on a large dataset of images of cracked reinforced concrete structures. The results showed that the model was able to accurately detect cracks of different types and sizes in the concrete structures. Yousefnezhad et al. (2020) proposed a deep learning framework for the automatic detection of cracks in building facades using an ensemble of convolutional neural networks. The proposed framework was able to accurately detect and classify a range of crack types. Awasthi et al. (2021) explored the use of deep learning techniques for the detection of cracks in concrete structures. The study focused on the use of transfer learning in training deep neural networks for crack detection in concrete pavements. The results showed that the proposed approach could accurately detect and classify different types of cracks in concrete structures. More recently, a study by Singh et al. (2021) proposed a deep learning-based algorithm for automated crack detection and segmentation in concrete structures. The study used a dataset of over 2,500 images of concrete surfaces with different types of cracks. The proposed algorithm achieved more than 97% accuracy in crack detection and segmentation.

In conclusion, the reviewed literature shows that deep learning techniques, particularly CNNs, have been applied successfully in detecting and characterizing cracks in buildings and other structures. The accuracy of these models can be improved by increasing the amount and quality of data used for training. Therefore, as more research in this area is conducted, we can anticipate more accurate and effective deep learning models for the detection of cracks in buildings. The focus of this work is to characterize the photographic cracks on various surfaces of Building using deep learning technique.

The objective of this project is limited to deep learning algorithms to investigate how cracks could be classified using deep learning algorithms.

i. Obtaining data sets for testing:
   In order to have a solid test set for the various models tested during the project, we need to create several different dataset of different cracks.
ii. Obtaining a data set for training:
   Finding many cracks is difficult, so we need to create some kind of data set or compensate for the lack of real crack images.
iii. Explore the possibilities of creating and implementing simulated datasets.
   Comparing different models with different designs to find reasonable candidate models as classifiers.

2. CLASSIFICATION OF CRACKS

2.1 Crack:

A crack is defined as complete or incomplete separation of concrete or masonry into two or more pieces caused by cracking or spalling. Cracks are classified as occurring in either plastic concrete or hardened concrete. The cause of any type of crack depends on a variety of factors and can be as simple as appearance or indicate serious structural damage or lack of durability. Cracks may represent the overall extent of damage or indicate a larger problem.

1.2 Types of cracks:

Cracks are classified into five types as given below

- Structural Cracks
- Concrete Cracks
- Plaster Cracks
- Joint Cracks
- Floor Cracks

2.2.1 Structural Cracks:

Cracks larger than 1/8 inch or 3 mm are considered structural cracks (Fig. 1). They usually appear as horizontal cracks, diagonal cracks or stepped cracks. They often have a nearly symmetrical pattern. Common causes of such crack are soggy ground after heavy rain, poor construction sites, planning errors, and ground movement. Soil shrinkage due to prolonged drought also triggers this type of cracking. When these cracks appear, windows and doors become difficult to close. You may also notice that the floor is starting to tilt. Structural issues can make your home uncomfortable. In addition to reducing structural integrity, cracks can affect indoor air quality, create conditions for mold growth, and create openings for vermin and crawling insects to enter your home.

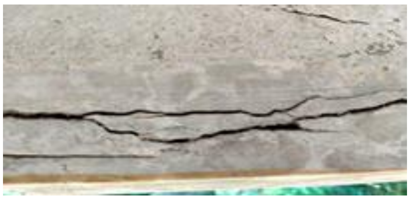

Fig -1: Structural Crack

2.2.2 Concrete Crack

Outdoor concrete pieces usually shrink during curing. This shrinkage is due to the evaporation of some of the water contained in the concrete. Cracking (Fig. 2) occurs when the shrinkage force exceeds the strength of the concrete. This next can be seen as a race against time between two phenomena. It evaporates water and increases the strength of concrete. This also applies to non-deformable parts.

Fig -2: Concrete Crack

2.2.3 Joint Crack

A construction joint (Fig. 3) is any joint built into a structure to be built to limit the number of possible cracks or to facilitate the construction process. A construction seam is an engineered crack and movement control point or is required for the construction project. Every material expands and contracts, so without construction joints, cracks and deformations would occur in the structure itself. In addition, the correct design of construction joints allows to build a project without procedures and techniques that add money and time to the project.

Fig -3: Joint Crack

2.2.4 Plaster Crack

Plaster cracks (Fig. 4) are small cracks in plaster walls. As surprising as it may sound, most cast cracks are very common and completely normal. As the gypsum dries and shrinks, most properties will show slight cracking. For newer buildings and properties that have been expanded relatively recently, these buildings may develop cracks in the walls as the foundation remains stable. It may come as a surprise, but it takes one to three years for a building to stabilize. During this time, the weight of the building and extensions causes the plaster to move and crack in places. Humidity, temperature changes, and humidity levels also cause similar cracks in plastered walls. Fluctuations in conditions cause the plaster to expand and contract, causing the entire building structure to swell or sag. This can cause fine cracks in the plaster that grow into more visible cracks over time. These fine cracks can also form when a freshly plastered wall dries.

Fig -4: Plaster Crack

2.2.5 Floor Crack

Cracked floors (Fig. 5) can be a sign of improper construction, uneven ground, and subsidence of the foundation due to the weight of the house. When it comes to incorporation settlements, the issue of incorporation can be devastating. Over time, cracks in floors can get worse and cost thousands of dollars to repair. This is also create other issues such as sagging floors, crooked walls, or cracked walls. Cracks in panels are usually thin and may look harmless, but they can be one of the biggest problems in your home. Your house may lean, your windows may stick together, your doors may become unbalanced, and puddles may form around cracks in your home. Most of the time, these buttress problems should be fixed, but if you have more than one of these problems, you most likely have a serious underlying problem.

Fig -5: Floor Crack

3. DEEP LEARNING & ITS APPLICATIONS

3.1 What is Deep Learning?

These neural networks attempt to emulate the behavior of the human brain, although far from matching its capabilities, allowing it to "learn" from large amounts of data. Although a neural network with one layer can still make rough predictions, additional hidden layers can help optimize and refine the accuracy. Deep learning drives a variety of artificial intelligence (AI) applications and services that enhance automation, performing analytical and physical tasks without human intervention. Deep learning underpins everyday products and services (such as digital assistants, voice-activated TV remotes, and credit card fraud detection) as well as new technologies floating (such as self-driving cars)

3.2 How Deep Learning Works?

Deep learning neural networks (artificial neural networks) attempt to mimic the human brain through a combination of data inputs, weights, and biases. These elements work together to accurately recognize, classify, and describe objects in your data. A deep neural network consists of multiple layers of interconnected nodes, each building on the previous layer to refine and optimize prediction or classification. This Propagation of computations through a network is called forward propagation. The input and output layers of deep neural network are called visible layers. The input layer is where deep learning models ingest data for processing, and the output layer is the final prediction or classification. Another process, called back propagation, uses an algorithm like gradient descent to calculate the error in the predictions and then go backwards through the layers to train the model, thus adjusting the weights and biases of the function. Using forward and back propagation together allows a neural network to make predictions and correct errors accordingly. Algorithms get more and more accurate as time goes on. The above describes the simplest type of deep neural network in its simplest terms. However, deep learning algorithms are very complex and there are many types of neural networks to tackle specific problems and data sets, for example,

Convolutional neural networks (CNNs) are primarily used in computer vision and image classification applications to detect features and patterns in images to enable tasks such as object

International Journal of Scientific Research in Engineering and Management (IJSREM)
Special Edition - Volume 09 ICITES -2025 SJIF Rating: 8.586 ISSN: 2582-3930

recognition. In 2015, CNN beat humans for the first time in an object recognition task.

Recurrent Neural Networks (RNNs) work with sequential or time-series data and are therefore commonly used in natural language and speech recognition applications.

3.3 Applications of Deep Learning

Real-world deep learning applications are part of our daily lives, but in most cases they are so well integrated into products and services that users are unaware that complex data processing is going on in the background. Few of the examples of these include:

- **Law and Order**

Deep learning algorithms can analyze and learn from transaction data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision, and other deep learning applications are extracting patterns and evidence from audio, video, images, and documents to improve the efficiency and effectiveness of investigative analysis, helping law enforcement to process vast amounts of data. Enable faster and more accurate analysis.

- **Finance**

Financial institutions routinely use predictive analytics to drive algorithmic stock trading, assess business risk on credit approvals, detect fraud, and manage client lending and investment portfolios.

- **Customer Care**

Many companies are integrating deep learning techniques into their customer service processes. Used in various applications, services, and customer service portals, chatbots are a simple form of AI. Traditional chatbots use natural language and visual recognition commonly found in menus like call centers. However, more sophisticated chatbot solutions use learning to try to determine if ambiguous questions have multiple answers. Based on the responses received, the chatbot either answers those questions directly or attempts to transfer the conversation to a human user. Virtual assistants such as Apple's Siri, Amazon Alexa, and Google Assistant extend the chatbot idea by enabling speech recognition capabilities. This creates new ways to target users in a personalized way.

- **Health Care**

The healthcare industry has benefited greatly from deep learning capabilities ever since hospital records and images were digitized. Image recognition application Empower medical imaging professionals and radiologists to analyze and evaluate more images in less time.

3.4 Artificial Neural Network

Neural networks are usually drawn parallel to the way the human brain processes information. Artificial Neural Networks (ANNs) often consist of interconnected networks of vast numbers of neurons. Each neuron in an artificial neural network is a mathematical function that receives one or more values, which are run through an activation function and passed as one or more values. A neural network usually has three parts: in, out, and an activation function. A mathematical model is shown in Figure 6.

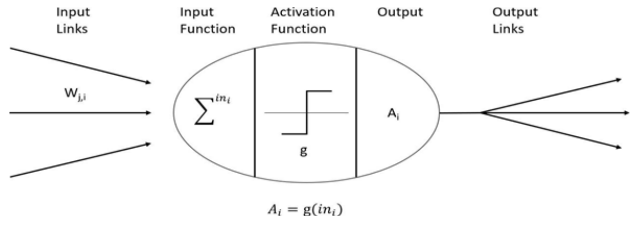

Fig -6: Mathematical model of artificial neuron
Image inspired by Octavian's blog

You can then modify the network to get the most accurate results possible for your chosen loss function, either by computing the gradient in the weight space or by simply back propagating the neural network. It's the neuron "weights" that actually change. Weights tell the neuron how much to use each input signal. They are shown to the left of the artificial neuron in Figure 6.

3.5 Convolutional Neural Network

Convolutional Neural Networks (CNNs), as the name suggests, are a type of deep neural network whose purpose is often to analyze images with some form of computation. Field of view. A CNN is a multilayer network that follows the typical structure of convolutional layers followed by subsampling layers, all finally connected with a dense layer to the final softmax vector. This type of design comes from the pioneering ConvNet, LeNet-5. LeNet-5 is a simple CNN for recognizing handwritten digits on a 28x28 grid. More advanced CNNs are often composed of multiple blocks, each of which has a type similar to this classical design, sometimes creating models hundreds of layers deep.

3.6 Textures

Texture classification has long been a difficult step in machine learning because textures are very different from object-based classification. Textures follow a set of statistical properties and regularly repeated text with some variation. Textures can be anything from fully stochastic patterns to fully regular patterns. Texture analysis is usually categorized into four sub-problems: classification, segmentation, synthesis, and shape. Only classification problems will be investigated in this project. Early research on texture features in the 1980s and 1990s focused on two areas: filtering approaches and statistical modeling. A filtering approach emphasizes texture features. They often use convolutional filters such as Gabor, pyramidal wavelets, or simpler filters such as Gaussian differencing. Statistical modeling characterizes textures resulting from probability distributions found in random fields. Since then, this area has been divided into three areas or methods of classifying textures: BoW, CNN, and attribute-based classification. Interesting information in this area of machine learning algorithms useful in this project comes from his report published by ICLR in 2019. In short, this report suggests that CNNs are consistently texture-biased, and also presents several results supporting this hypothesis. Additionally, this report highlights two interesting studies that suggest that texture information alone is sufficient for many ImageNet CNNs to classify objects. The first by Gatys et al. We found that the linear classifier on the CNN's texture map showed little change in loss compared to the original network. In a second report, Brendel & Bethge showed how a CCN with a very small incoming field penetrates all layers. Achieving high accuracy in ImageNet means that the CNN can accurately classify objects without having much feature or shape information. As the report suggests, this is not the case unless the CNN is particularly good at detecting local patches of texture that provide enough information to classify the object. These types of properties make CNNs very interesting candidates for crack classification problems.

4. MODEL DEVELOPMENT USING SOFTWARE TOOLS

4.1 Software Tools

Several tools were used during the experimental work. In order to focus on the testing and data collection of various models, no major background research has been done on the various complex features used by many tools and is not detailed in this report. This section briefly describes all the different tools used during the experimental work. More interested readers should refer to the respective sources.

- **Tensor flow**

Tensor flow is an machine learning platform. It includes a large ecosystem of tools, libraries, and common sources that make it easy to create machine learning algorithms without knowing all the advanced and difficult features. Tensorflow makes it easy to experiment with different structures and parameters, and easily train models on GPUs to significantly reduce training time

- **Keras**

Keras is a high-level neural network application programming interface that runs on top of Tensorflow, CNTK, or Theano. It is designed to make machine learning experimentation easier and move from idea to result faster. Keras models are built in a sequential format, and their simple coding design makes it easier to work with additional functions, modules, loss functions, and activation functions. The model is written in Python, which makes the code compact and easy to debug and develop. The Keras API also provides usage of pre-trained models on ImageNet. It can be used with or without a classification layer.

- **Blender**

Blender is a free 3D modeling, 3D graphics, animation, game design, effects and rendering tool. This is a public project developed with the support of the Blender Foundation and used by everyone from students to After Effects professionals. Create a cracked wall simulation using Blender to keep the simulation within a classifiable range while allowing you to create large data collections in different environments.

4.2 Crack Detection Model Development

4.2.1 Data Collection

The size and content of your dataset can greatly affect your model's performance. Smaller images result in smaller data sizes and faster training speeds. On the other hand, images that are too small do not contain enough data. The default input size for the VGG16 architecture is 224 x 224, but can accommodate slightly larger images. A classification building crack image dataset was used for this study. This data set contains 5000 images with 227 x 227 pixels in RGB channels. The image is placed in the positive (crack) and negative (crack) category. Each category contains 1000 images.

4.2.2 Image Processing

The images were individually processed using TensorFlow after import to transform the image into a series of elements. I then used the Sci-Kit Image package in Python 3.8 with separate code to process the image. IP crack detection uses various image pre-processing techniques to enhance crack features. DL methods typically do not use image pre-processing techniques to reduce the amount of data the algorithm needs to learn. I processed the images to get four datasets.

4.2.3 Proposed CNN Model

The current investigation was performed using a pre-trained CNN architecture. Transfer learning using pretrained architectures has been proven to improve the efficiency and accuracy of crack classifiers. Moreover, VGG16 is the most widely used pre-trained architecture for crack detection and ImageNet. The proposed CNN architecture consists of convolutional blocks and fully connected layers. Each convolutional block consists of a convolutional layer, an activation unit, and a pooling layer. A convolution layer uses a kernel / filter to perform a convolution operation on the output of the previous layer.

4.2.4 Model Development

The calculations were performed on a laptop (RAM: 16GB, CPU: AMD Ryzen 7 4800H, Radeon Graphics 2.90 GHz, GPU:Develop models on NVIDIA GeForce GTX 1650 Ti, 64-bit operating system). The code was programmed using Anaconda Spyder 4.1.5 (Python 3.8) with Python packages Keras 2.4 and TensorFlow 2.5. This allowed me to run CNN on any system using Python and a suitable PIP or Python notebook. The Keras package contains so many DL architectures. The Keras VGG16 model is based on the model proposed by Simonyan and Zisserman (2014). Max pooling was used for best crack detection results. The model was trained using the Adam optimizer and the binary cross-entropy of the loss. A training period is also specified. An epoch uses all images to train the model and each step is a batch. A recommended batch size of 200 images was used. 50 epochs were used to train the images. An input layer defines the input parameters of the model. Position, label, number of channels, dataset name, size (pixels). Sequential shift is data augmentation. We expanded the data with a rotation ($0.2/2\pi$) and flipped it horizontally and vertically to increase the data size and remove orientation errors. Finally, a normalization layer normalizes the data (0 to 255) so that ImageNet weights can be used. The ImageNet weights used for pretraining range from 1 to +1. Then the VGG16 architecture was used in pre-training. The pooling layer converted the original shape of the base model into vectors. The dropout feature turns filters off during exercise to customize your results. Finally, a dense classification layer combined the model values into binary results. This result was used to predict crack designation. Cracks were marked 0 and cracks were marked 1. The program was rounded to predict cracks. A function was created that converts the number 0.5 to 1. This allowed the program to determine if the image was cracked.

4.2.5 Model Analysis

Analysis of the model evaluated the classifiers for their ability to predict the classes. In both cases, the data had a class designation of either positive or negative (p, n). The classification model predicted the class as Yes or No (Y, N). A classifier with 2 distinct classes produced 4 results per instance False-positive results refer to values incorrectly predicted to be positive in reality. H. Positive predicted negative values, false negative values associated with negative predicted positive values; true positive and true negative values are correctly predicted values. Classification performance was evaluated using the following five criteria: Accuracy (ACC), true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), negative predictive value (NPV), and F1 score. The F1 score is the harmonic mean of the data. All formulas were created with the same four values. True positive (TP), true negative (TN), false positive (FP), and false negative (FN). This allowed us to properly evaluate the CNN based on the classification. Recall (TPR and TNR) is a CNN feature that finds all relevant categorical cases in the data. Accuracy (PPV and NPV) is the ability of a CNN to identify only the correct categories. The F1 value is preferred because it is the harmonic mean of the two and thus reduces the impact of extreme results in terms of precision or recall. All model results were compared to the control (RGB) to determine if there was any improvement

5. RESULTS AND DISCUSSION

Different CNN models using transfer learning was built and generated 5 models (5 different images in 2 training rounds). The model generated a confusion matrix that allows comparison and evaluation of performance. This section describes the confusion matrix results generated from the test data (5000 sample images) and the derived metrics. Validation data were used for model training. The model training accuracy was plotted at each epoch to visualize the training progress and show the improvement of the model during training. RGB data was chosen as a control for testing purposes. However, the pretrained VGG16 model only had access to the pretrained RGB weights. The RGB weights were designed for three channels (red, green, blue), but the processed image had only one channel (brightness). Therefore, depending on the model's color dependencies, RGB results were expected to improve performance.

5.1 50-EPOCHS Training

50 epochs of training was performed to determine whether larger training could improve the model's correction of single-channel images. Confusion matrix results show that the 50 epoch model performs better for all models.

Fig -7: 50-Epochs Loss

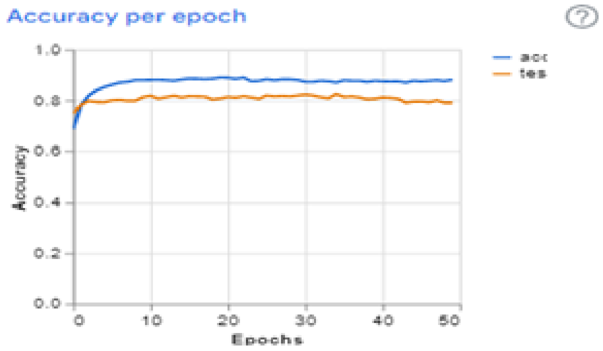

Fig -8: 50-Epochs Accuracy

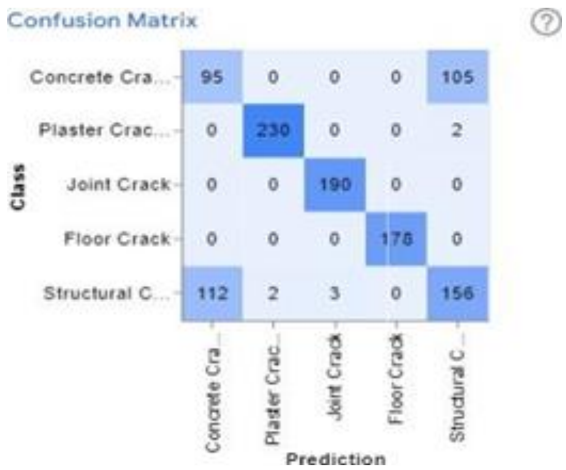

Fig -9: 50-Epochs Confusion Matrix

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 -10: 50-Epochs Accuracy per Class

5.2 Structural Crack Detection Results

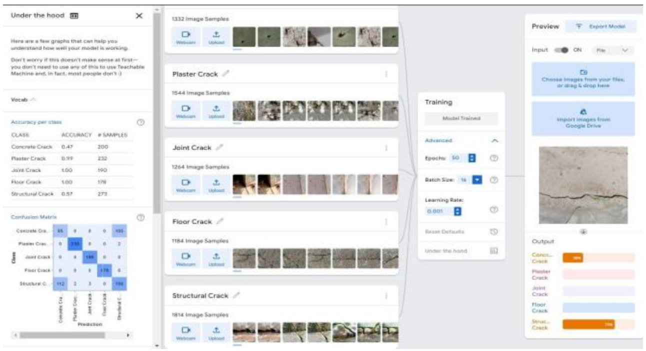

Fig -11: Structural Crack Detection Results -1

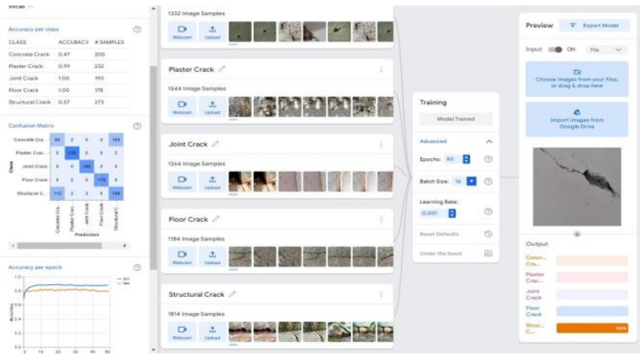

Fig -12: Structural Crack Detection Results -2

5.3 Concrete Crack Detection Results

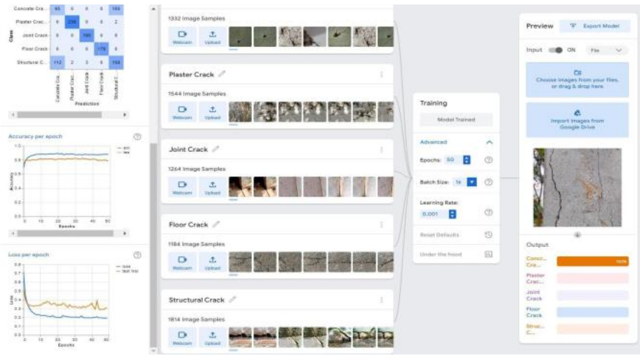

Fig -13: Concrete Crack Detection Results -1

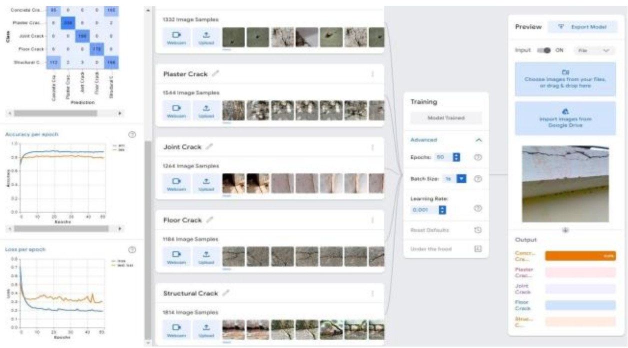

Fig -14: Concrete Crack Detection Results -2

5.4 Floor Crack Detection Results

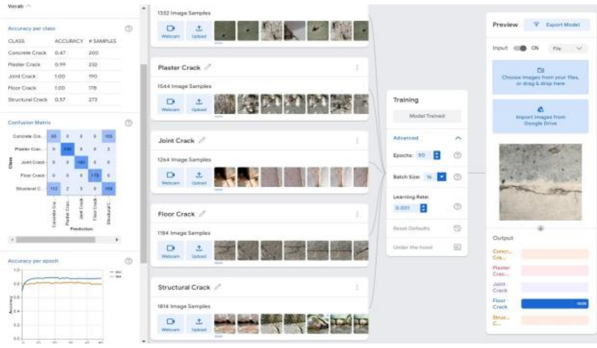

Fig -15: Floor Crack Detection Results -1

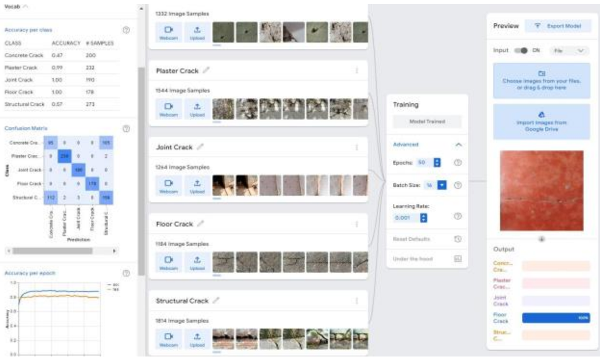

Fig -16: Floor Crack Detection Results -2

5.5 Joint Crack Detection Results

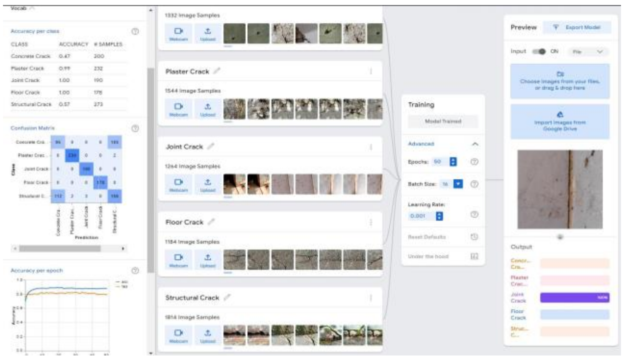

Fig -17: Joint Crack Detection Results -1

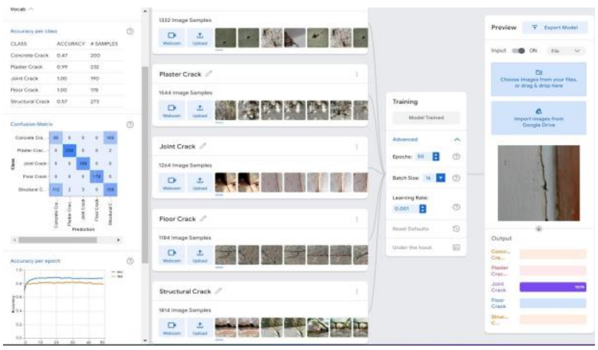

Fig -18: Joint Crack Detection Results -2

5.6 Plaster Crack Detection Results

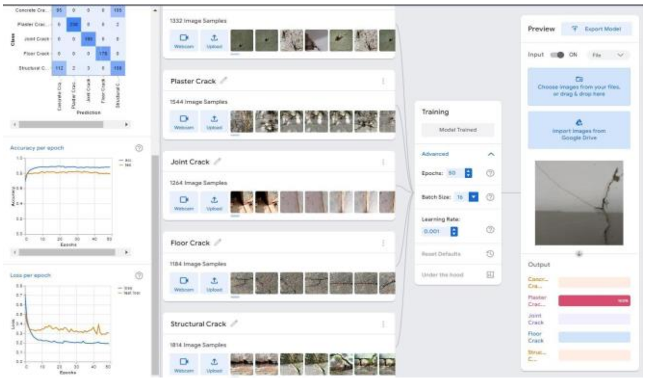

Fig -19: Plaster Crack Detection Results -1

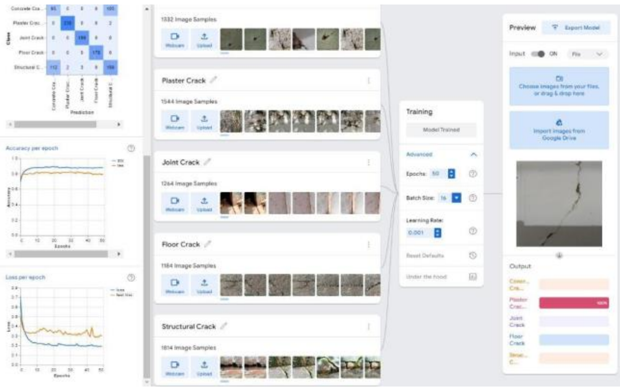

Fig -20: Plaster Crack Detection Results -2

6. CONCLUSIONS

- This study investigated the effect of image pre-processing on the performance of DL crack detection using a data set of 5000 images. The results showed that using a pretrained model with RGB weights does not affect the performance of the CNN model for detecting cracks in concrete structures. In this study, we developed a CNN using the Keras Python package and pretrained VGG16. The original image data set was converted into five comparison sets using the SciKit Image Python package.
- Model (F1 score for 50 epochs: 99.549%) showed similar performance to the RGB model (F1 values for 50 epochs: 99.533%). This suggests that the features used in DL to identify cracks are color independent. The model performed very well in terms of accuracy and F1 values (all above 98%)
- This study showed that color was not a relevant feature for DL crack detection. This was promising because color images are getting larger, and the smaller image data size can improve processing speed and potentially reduce the data size required for storage. These results can be misleading due to the pre-trained model's RGB weights. However, further studies should examine these results using fully trained and segmented models. Tests, results and analyzes were performed on pre-trained models. Pre-trained model weights are only available in 3 channels (RGB). This was done due to lack of time and the more knowledge required to do it. We need to receive the weights of one channel. Further investigation can be done into the impact of IP-FCN pixel segmentation on improving pixel accuracy. Additionally, you can explore other pretrained models such as VGC11, VGC19 and AlexNet and compare their performance.

ACKNOWLEDGEMENT

The authors thank the Department of Civil Engineering, Karunya Institute of Technology & Sciences (KITS) and Department of Computer Science and Technology, KITS for providing access to all the resources used for our study.

REFERENCES

1 1. Kim, G., Jang, J., Kim, K., & Lee, S: Automatic pavement crack detection using an adaptive threshold method and a deep belief network. IEEE Transactions on Intelligent Transportation Systems, (2011) 12(3), 808-817. doi:10.1109/tits.2011.2144618

2 Ouyang, W., Zhang, X., Xie, C., & Huang, Q: Pavement crack detection using deep convolutional neural networks. In Proceedings of the 2013 IEEE International Conference on Image Processing (ICIP) (2013) (pp. 285- 289). doi:10.1109/icip.2013.6738085

3 Kim, L., Park, D., Lee, D., Lee, J., & Park, S: Automatic detection of surface cracks in concrete structures using a deep learning-based approach. Computer-Aided Civil and Infrastructure Engineering, (2013) 28(8), 604-618. doi:10.1111/mice.12033

4 Moon, H. & Kim, J: Intelligent crack detecting algorithm on the concrete crack image using neural network, in Proceedings of the 28th ISARC, Seoul, Korea, (2011) 1461–67

5 B. Y. Lee, Y. Y. Kim, S.-T. Yi, and J.-K. Kim: “Automated image processing technique for detecting and analysing concrete surface cracks,” Structure and Infrastructure Engineering, (2013) vol. 9, no. 6, pp. 567–577.

6 Ouyang W, Xu B: Pavement cracking measurements using 3D laser-scan images. Measurement Science and Technology (2013) 24:105204

7 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,” *Journal of Signal Processing Systems*, (2014) vol. 77, no. 3, pp. 221–240.

8 Peng Wang, Ananya, Ruqiang Yan, RobertX. Gao: Virtualization and deep recognition for system fault classification, Journal of Manufacturing Systems Volume 44, Part 2, (2017) Pages 310-316

9 Bi, Y., Xu, L., & Fan, C: Crack detection based on deep convolutional neural network. Journal of Physics: Conference Series, (2017) 908, 012074. doi:10.1088/1742-6596/908/1/012074

10 Wang, X., Yao, W., Wu, Z., & Wang, Y: Pavement crack detection using a deep learning-based approach. Journal of Computing in Civil Engineering, (2017) 31(4), 04017034. doi:10.1061/(ASCE)CP.1943-5487.0000669

11 Zhang, C., Li, J., Song, Y., & Zhang, L: Multi-feature learning and multi-scale fusion for crack detection using deep convolutional neural network. Sensors, (2018) 18(6), 1793. doi:10.3390/s18061793

12 Liu, B., Yao, Q., Li, W., & Yang, Z: Crack detection in building facades using an attention-based convolutional neural network. Remote Sensing, (2018) 10(5), 678. doi:10.3390/rs10050678

13 Salleh, N. M. M., Yusop, Z., Mohd, M. J., Mansor, W., & Abdul-Wahab, M. F: Deep learning approach of crack detection in concrete structures. IOP Conference Series: Materials Science and Engineering, (2020) 737, 012061. doi:10.1088/1757-899X/737/1/012061

14 Awasthi, A.K., Koduru, J., Arora, H., & Kumar, R: Crack detection in concrete structures using deep learning techniques: a review. Journal of Civil Structural Health Monitoring, (2021) 11, 307–332. doi:10.1007/s13349-020-00436-x
15 Huang, H., Hao, Q., & Zhang, S: Crack detection of road pavement using deep learning methods. Measurement, (2019) 137, 581-592. doi:10.1016/j.measurement.2018.12.025
16 Ouali, A., Hanini, S., Zerrouki, A., & Nouioua, F: Crack detection in reinforced concrete structures using deep learning. Journal of Civil Structural Health Monitoring, (2020) 10, 935–951. doi:10.1007/s13349-020-00406-3
17 Yousefnezhad, M., Hasanipanah, M., & Ramezani, M: A deep learning ensemble framework for automatic detection of cracks in buildings facades. Neural Computing and Applications, (2020) 32, 7983-7997. doi:10.1007/s00521-019-04495-4
18 Singh, R. K., Abhay, N. D., Dey, N., & Bhatt, C: CrackDetNet: A convolutional neural network-based approach for automated crack detection and segmentation in concrete structures. Automation in Construction, (2021) 126, 103684. doi:10.1016/j.autcon.2021.103684