DAMAGE DETECTION IN CONCRETE STRUCTURES USING MACHINE LEARNING TECHNIQUE

Project Reference No.: *47S_BE_2748*

College ***:*** *Vidyavardhaka College of Engineering, Mysuru*
Branch ***:*** *Department of Civil Engineering*
Guide(s) ***:*** *Dr. Rajeeth T. J & Dr. Varshitha D. N*
Student(S) ***:*** *Ms. Monika R*
Ms. Sowprathi P
Ms. Padma Norzin
Mr. Moham med Abbas

Keywords:

Damage detection, Deep learning, Convolutional Neural Network, VGG-16 Network, Concrete Structures

Introduction:

In today's world, the integrity and durability of infrastructure are pivotal for societal functionality and economic prosperity. Infrastructure, encompassing buildings, bridges, roads, and dams, serves as the backbone of communities, necessitating continuous monitoring and maintenance to ensure safety and sustainability.

The main concern is the detection of cracks within these structures, often arising from age, environmental factors, or structural stresses, which, if left undetected, can compromise structural integrity, leading to safety hazards. Traditional manual crack detection methods are labour-intensive, time-consuming, and error-prone, underscoring the need for more efficient and accurate solutions. The AI and Machine Learning tools is one of the pioneer techniques to address these challenges.

The ongoing work mainly focuses on leveraging deep learning models for automated crack detection in concrete structures, particularly walls and slabs. The project utilizes a comprehensive dataset sourced from diverse repositories, including online databases, industry sources, and locally collected data. By integrating deep learning technologies Convolutional Neural Network (CNN) and with Visual Geometry Group-16 (VGG-16) model the works aims to streamline crack detection processes, enhance accuracy, and facilitate proactive interventions.

Objectives:

1. To collect the data related to the cracks/damages in concrete structures.
2. To develop a machine learning model capable of detection and assessment of structural damages in concrete structures.
3. To reduce the computational complexity involved in detection and assessment of damages in concrete structures.

Methodology:

- In the data collection phase, a diverse dataset was collected from various sources, which includes online sources, industry databases, and also data was collected locally around the college premises.
- Additionally, each image was categorized and annotated with its corresponding crack type, and data of each type of cracks were stored in google drive

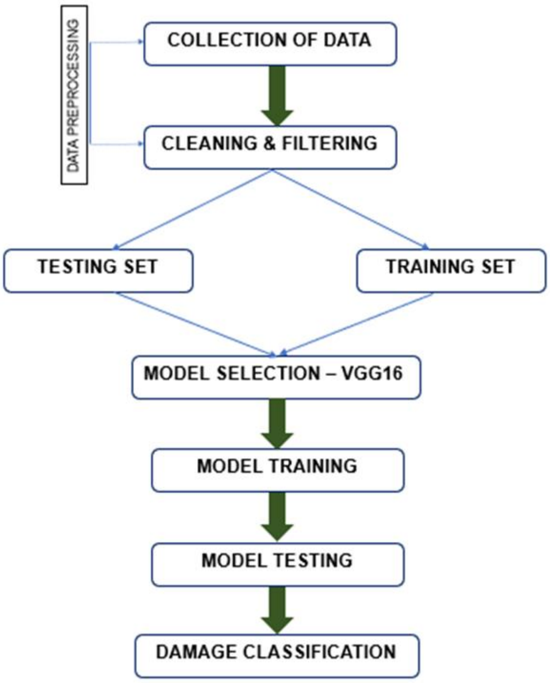

Figure 1: Methodology Followed

Model Used:

- After conducting in-depth research and considering factors such as the nature of the data, the specific task requirements, and the dataset size, it was concluded that the VGG16 model is an optimal choice.
- To train the VGG-16 model, Python language is used in Jupyter Notebook, were Jupyter Notebook was launched in Anaconda Navigator.
- Model training is start by importing necessary libraries for implementing VGG-16 using the sequential method, where layers are arranged in sequence.
- Model is initialized by setting up a convolutional neural network (CNN) using the VGG16 architecture for image classification tasks. Initially, the image size is specified as 100x100 pixels, and key parameters such as epochs (set to 5), batch size (set to 32), and file paths for training and validation data are defined.
- The compilation of a neural network model is done using Kera's and also specified the loss function as categorical cross-entropy, the optimizer as RMSprop, and includes accuracy as a metric for evaluation during training.
- The training set comprising around 70% of the data, is used to train the model and the validation set, which constitutes the remaining 30% of the data, allows for

unbiased assessment of the model's accuracy, precision, recall, and other performance metrics on new, unseen examples.

- With all essential input data and configurations provided, the model is ready to run, utilizing the training and validation sets, as well as the specified number of epochs and batch size.

Results:

Model testing

- To test the trained model, it's crucial to use a crack photo not included in the training or validation sets, ensuring the model's evaluation on unseen data. The image format must be JPG or JPEG for the model to predict the crack type accurately.

- This testing process evaluates the model's performance on unseen data, providing insights into its accuracy and effectiveness in detecting cracks in concrete structures.

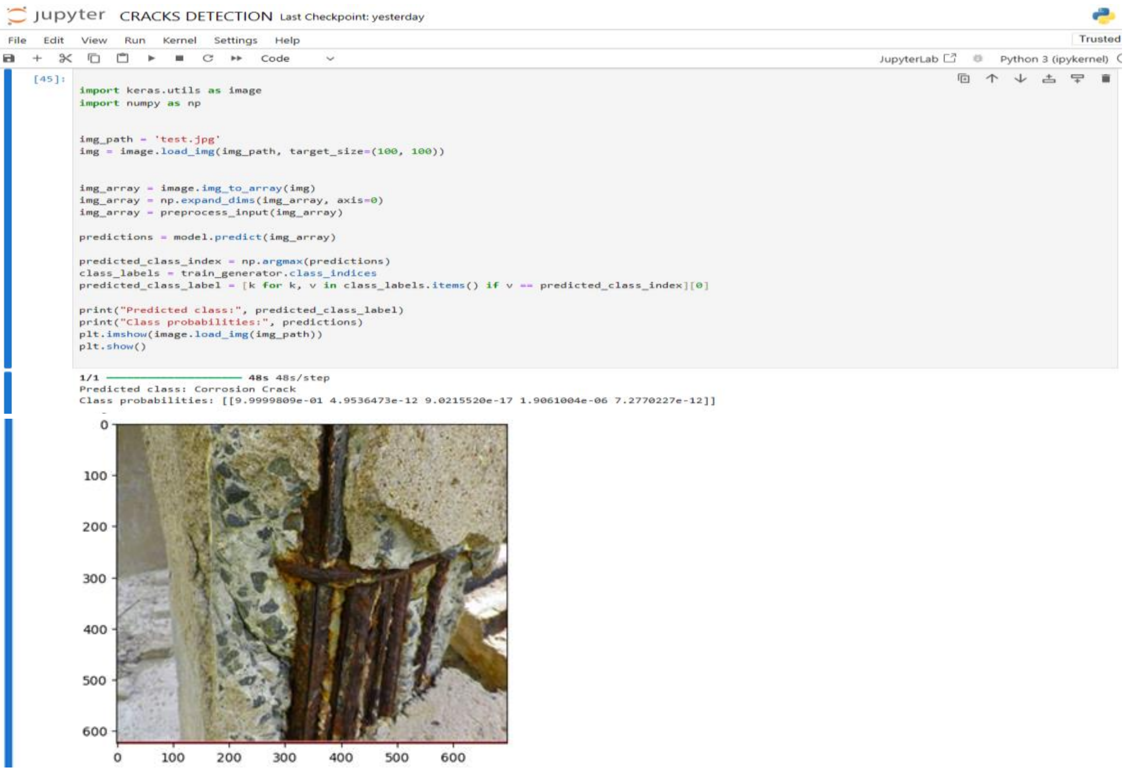

Figure 2: Testing the model

Model summary

- After model running it shows accuracy and loss metrics are reported for both the training and validation datasets across multiple epochs. As the training progresses, there's a noticeable trend of increasing accuracy and decreasing loss, indicating that the model is learning and refining its predictive capabilities over time.

- Our model has achieved maximum of 83% accuracy which indicates that the model is performing reasonably well in crack classification task. It demonstrates that the model is making correct predictions for the majority of instances, which is a positive sign of its effectiveness.

- Overall, an accuracy of eighty-three percent is a good outcome, but further analysis and evaluation are recommended to ensure the model meets the desired performance standards for its intended application.

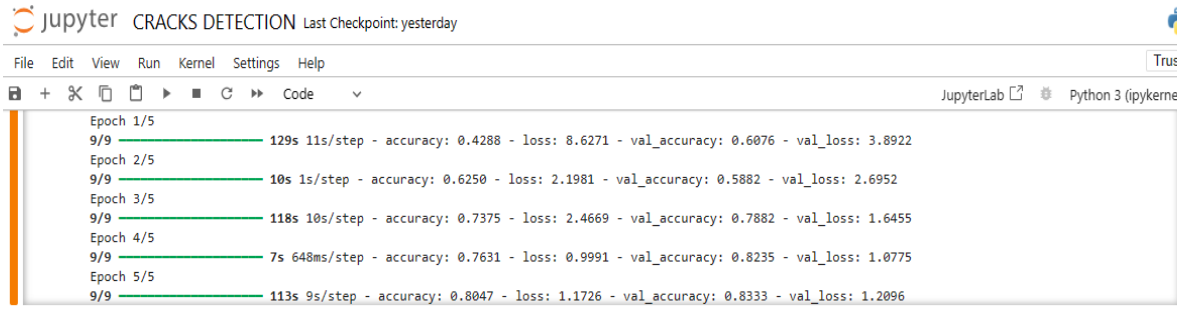

Figure 3: Model accuracy

Model: "functional_15"

Layer (type)	Output Shape	Param #
input_layer_7 (InputLayer)	(None, 100, 100, 3)	0
block1_conv1 (Conv2D)	(None, 100, 100, 64)	1,792
block1_conv2 (Conv2D)	(None, 100, 100, 64)	36,928
block1_pool (MaxPooling2D)	(None, 50, 50, 64)	0
block2_conv1 (Conv2D)	(None, 50, 50, 128)	73,856
block2_conv2 (Conv2D)	(None, 50, 50, 128)	147,584
block2_pool (MaxPooling2D)	(None, 25, 25, 128)	0
block3_conv1 (Conv2D)	(None, 25, 25, 256)	295,168
block3_conv2 (Conv2D)	(None, 25, 25, 256)	590,080
block3_conv3 (Conv2D)	(None, 25, 25, 256)	590,080
block3_pool (MaxPooling2D)	(None, 12, 12, 256)	0
block4_conv1 (Conv2D)	(None, 12, 12, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 12, 12, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 12, 12, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 6, 6, 512)	0
block5_conv1 (Conv2D)	(None, 6, 6, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 6, 6, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 6, 6, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 3, 3, 512)	0
flatten_7 (Flatten)	(None, 4608)	0
dense_7 (Dense)	(None, 5)	23,045

Total params: 14,737,733 (56.22 MB)
Trainable params: 23,045 (90.02 KB)
Non-trainable params: 14,714,688 (56.13 MB)

Figure 4: Model summary

Conclusion:

- The utilization of the VGG-16 model significantly improves the accuracy of damage detection in concrete structures, ensuring early identification and prevention of potential structural issues.

- By accurately identifying damage in concrete structures, the model reduces the necessity for continuous monitoring, thereby easing the maintenance burden.

- The successful application of the VGG-16 model demonstrates the efficacy of advanced machine learning approaches in addressing complex challenges within the construction and infrastructure sector.

- These outcomes pave the way for innovation in structural health monitoring, fostering the adoption of advanced technologies for safer and more resilient infrastructure.

- Based on the trained model developed and the output seen, it can be declared that the VGG-16 model can predict the cracks and its types with high accuracy.

- Ultimately, the project contributes to the maintenance of safer and more resilient infrastructure by enabling early detection and identification of remedial measures of structural damage.

Innovation in the project:

Compared to traditional methods, this project innovates by leveraging the VGG-16 model's deep learning capabilities to accurately detect and classify damage in concrete structures, enhancing infrastructure safety and maintenance practices.

Scope for future work:

- Further refinement of the model to detect subtle or early-stage damage.
- Integration with real-time monitoring systems to facilitate proactive maintenance.
- Exploration of additional deep learning architectures for multi-model data fusion.
- Scaling up for application in broader infrastructure contexts beyond concrete structures.