ESP IJAST
ESP International Journal of Advancements in Science & Technology
ISSN: 2583-9233 / Volume 1 Issue 3 December 2023 / Page No: 20-26
Paper Id: IJAST-V1I3P104 / Doi: 10.56472/25839233/IJAST-V1I3P104

Original Article

Automatic Damage Detection of Historic Masonry Buildings Based on Densenet Deep Learning Model

Sheetal Sharma
Coleman University, San Diego Campus, USA.

Received Date: 03 October 2023 Revised Date: 10 November 2023 Accepted Date: 11 December 2023

Abstract: *Time-consuming and labor-intensive, vision-based manual inspection equipment is required to discover and assess superficial damage to historic structures. Currently, professional equipment and in-situ visual assessment procedures are used to detect surface degradation at historic sites. While there are clear benefits to using this technique in practice, inspector inexperience can lead to incorrect damage assessments that have lasting consequences for structural safety evaluations and repairs. This kind of examination is inefficient and time-consuming when applied to a large scale. It takes a lot of time and effort, as well as a large number of skilled individuals, to process and analyze the data afterwards. As a result, the requirements of fast damage identification in ancient structures are not addressed by this approach. A unique automatic damage detection approach was suggested in this study that uses a Mask R-CNN model built on the DenseNet framework to identify two types of damage (efflorescence and spalling) in old masonry buildings, therefore overcoming these limitations. Model validation in MATLAB Simulink for suggested flaw detection.*

Keywords: *Deep Learning Model, Damage Detection, Masonry Buildings.*

I. INTRODUCTION

Cracks in engineering structures, such as concrete surfaces and beams, often start at the microscopic level on the surface due to fatigue stress and cyclic loading. Cracks in the structure lead to material discontinuities and a decrease in local stiffness. Damage and potential failure can be avoided if problems are identified and addressed in a timely fashion. To identify a fracture in a structure using any processing approach is known as crack detection. There are two approaches of identifying cracks. There are two types of testing: destructive and nondestructive. Surface condition issues are assessed using eye inspection and surveying techniques.

Cracks on the surface of a structure can reveal information about its age, state of deterioration, and carrying capacity by virtue of their kind, number, breadth, and length.

Automatic crack detection is being developed to replace the time-consuming and subjective old human inspection processes with a faster and more reliable alternative for analyzing surface defects. As a result, a more secure way of surveying has been used. When it comes to non-destructive testing, automatic crack detection is superior. It is challenging to make an impartial assessment of degradation by manual inspection. Infrared and thermal testing, ultrasonic testing, laser testing, and radiographic testing are all examples of non-destructive testing methods that may be used for automated crack identification.

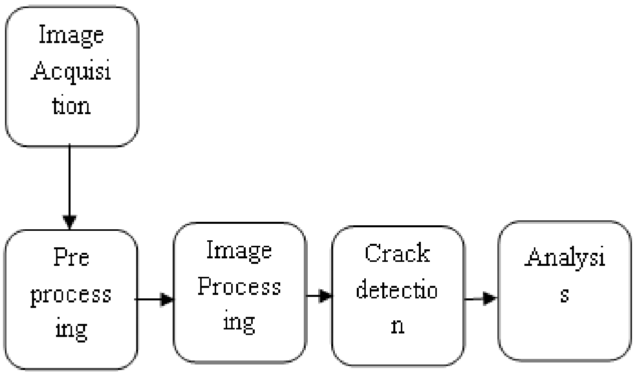


Figure 1: The Architecture of Image Processing Based Crack Detection

Image-based fracture detection for non-destructive inspection is gaining popularity. The uneven shape and size of cracks, as well as other issues like inconsistent lighting, shade, imperfections, and concrete spall, contribute to the challenges of image-based detection. Many of the detection approaches in image processing were proposed because of the ease with which they might be implemented. Integrating an algorithm, a morphological strategy, a percolation-based approach, and a practical methodology are the four broad categories into which these strategies fall.

The overall architecture for image-processing-based fracture detection is depicted in Fig. 1. Following are the procedures involved in the image processing method: (1) Using a camera or other means, take a first picture of the building that will be analyzed for cracks. (2) Following picture capture, the gathered images are subjected to pre-processing, which includes approaches like segmentation, in order to prepare them for the image processing operation. (3) The subtracted picture sample is processed using various methods in image processing. Here, utilizing the processed picture as a guide, (4) fracture detection on the structure will become apparent. In step five, crack feature extraction, the cracks are classified according to their diameter, depth, and propagation direction.

II. RELATED WORK

To classify cracks in reinforced concrete buildings as moderate, severe, or extremely severe using an Android app, A. Villanueva et al. intend to develop a deep learning model with Yolov3. Overall, the android app has a kappa value of.97 and a summary accuracy rate of 93.33 percent. Therefore, the deep learning model and android app accurately calculated the crack's detection and categorization.

The break in the surface of the infrastructure may be detected using an image processing technique and a portable system, as proposed by S. -M. Kang et al. This research use the camera-based approach, one of several available for detecting cracks. In this study, we deploy the system on real roads and evaluate its performance under those settings.

Pavement cracks are the most prevalent kind of road damage, hence E. Buza et al. proposed a novel unsupervised image processing approach for segmenting them. First, a picture is segmented into M N sub-images, and those without cracks are discarded according to an empirically determined threshold. Then, analysis is performed on a very small number of sub-images, drastically cutting down on processing time.

In this research, H. Yao et al. offer an updated version of the You Only Look Once (5th edition, or YOLOv5) algorithm for detecting pavement cracks. In order to identify cracks, a total of 12 attention models were constructed using the spatial and channel squeeze and excitation (SCSE) module and the convolutional block attention module (CBAM).

In order to identify pavement cracks in photographs, Y. Yu et al. create a unique context-augmented capsule feature pyramid network (CCapFPN). High-level, inherent, and prominent aspects of cracks are represented using vectorial capsules in the CCapFPN. The CCapFPN achieves its high-resolution, semantically robust feature representation for effective crack detection by employing a feature pyramid architecture to integrate capsule characteristics across several levels and scales. Results from quantitative analyses indicate a level of performance indicative of an accuracy of 0.9200, recall of 0.9149, and an F-score of 0.9174,

After learning high-level characteristics for crack representation, Q. Zou et al. offer DeepCrack, a fully trainable deep convolutional neural network for automated crack identification. To do so, our approach combines deep convolutional features learnt at different scales in a hierarchy to detect and characterize line formations.

J. M. Z. Maningo et al. design a crack detection system that can map the wall's surface and analyze the physical properties of cracks. Faster R-CNN machine learning architecture was utilized to train the model that would be used to categorize and identify fractures.

A semi-supervised technique for fracture identification in pavement was proposed by G. Li, J. Wan, et al. To begin, our suggested technique may make use of unlabeled pavement photos for model training, and the resulting model can provide a supervisory signal for those photographs that lack annotation.

You Look Only Once-version3 (YOLO-v3) is one of the most advanced deep learning models, and P. Kumar et al. employ the edge computing concept to present a real-time multi-drone damage detection system for tall civil structures. The suggested system is installed on Pixhawk's open source hexacopter and leverages Jetson-TX2 as the hardware platform on which YOLO-v3 is operated.

Using a combination of deep learning and image processing, S. B. Ali et al. want to create a fully automated crack inspection system. Convolutional neural network (CNN) transfer learning models are used to automatically classify surface photos into cracked and uncracked categories by learning the inherent properties of cracks from the surface images.

Crack detection was analyzed by S. Ogawa et al. from the perspective of picture segmentation. Using a mixture of Gaussian mixture models (GMMs) and filtering techniques from image processing, this research suggests a novel approach to crack detection. The experimental findings outperform the state-of-the-art crack detection approaches in terms of both accuracy and processing time, demonstrating the superiority of our suggested method.

Cracks in pipelines can be detected using an image processing technique that W. A. Altabey and colleagues suggested. Cracks in high-resolution crack pictures are recovered using a pipeline image segmentation model developed utilizing semantic segmentation.

Combining the illumination homogenization methodology with the deep convolutional neural network approach, D. Xu et al. offer a road fracture detecting algorithm. The Dice similarity coefficient of crack identification is enhanced to 0.7937, demonstrating that our suggested technique may increase the accuracy of crack detection in real-world road photos.

III. PROPOSED SYSTEM

Management of infrastructure is a crucial yet expensive endeavor. One of the most important aspects of managing infrastructure is repairing the fracture. Significant damage is done by large cracks in infrastructure, and their lifespan is shortened as a result.

Numerous computer-vision use cases focus on cracks because they are common line topologies. In order to identify two types of damage (efflorescence and spalling) in old masonry buildings, this study developed a unique automatic damage detection approach utilizing a Mask R-CNN model built on the DenseNet framework. Model validation in MATLAB Simulink for suggested flaw detection.

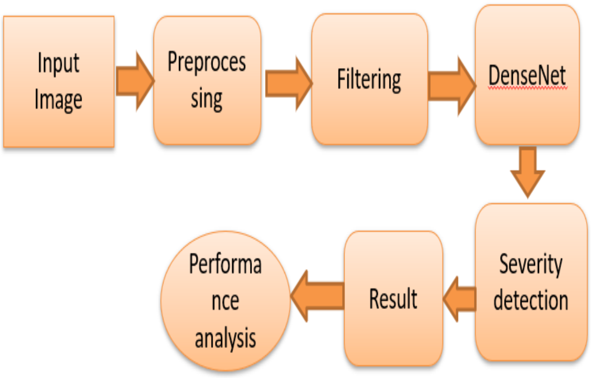


Figure 2: Proposed Technique

Each convolutional layer of a typical feed-forward Convolutional Neural Network (CNN) gets the output of the preceding convolutional layer and uses that information to generate an output feature map that is then passed on to the next convolutional layer. This means that if there are 'L' layers, then there will be 'L' direct connections, one from each layer to the next.

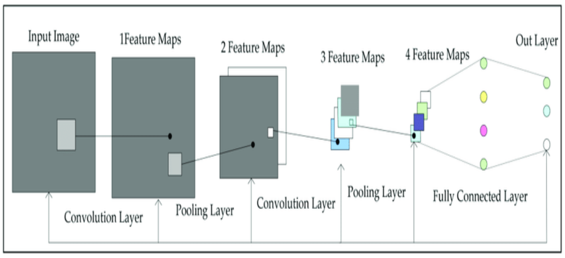


Figure 3: Conventional CNN

However, the 'vanishing gradient' problem emerges when the number of layers in the CNN increases, i.e. as they go deeper. Meaning that some information might 'vanish' or get lost when the path for information from the input to the output layers grows, reducing the network's capacity to train properly.

A. Mask R-CNN Architecture:

In 2017, the concept of Mask R-CNN was presented by Kaiming He et al. It's extremely similar to Faster R-CNN, with the addition of a layer specifically to provide segmented predictions. The second step, which generates region proposals in both architectures in parallel, predicts classes, makes bounding boxes, and produces binary masks for each region of interest.

DenseNets fix this issue by streamlining the connection pattern between layers of a typical CNN design. Densely Connected Convolutional Network (DenseNet) describes a network design in which every layer communicates directly with every other layer. For 'L' layers, the number of direct connections is L(L+1)/2.

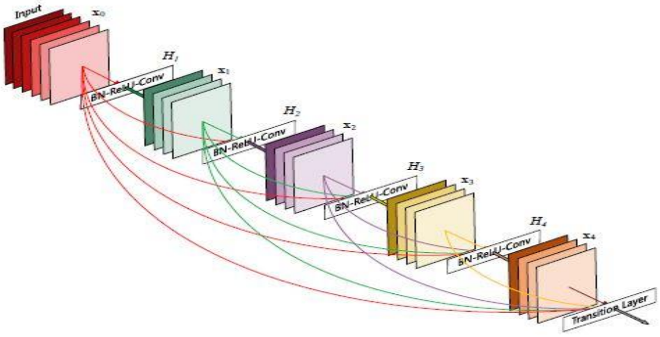


Figure 4: DenseNet Architecture & Components

Components of DenseNet include:

- Connectivity
- DenseBlocks
- Growth Rate
- Bottleneck layers

B. Connectivity

The feature maps from the preceding layers are not added together, but rather concatenated, and utilized as inputs in the current layer. Because of this, DenseNets may reuse features while still requiring fewer parameters than a conventional CNN of similar performance. In this setup, each successive layer (x0,..., xl-1) feeds its feature-maps onto the current layer (l):

$$\mathbf{X_l = H_l([X_0, X_1,....,X_{l-1}])}$$

When the feature-maps are concatenated into [x0,x1,...,xl-1], where [x0] is the output of layer 0 and [x1] is the output of layer 1, and so on. Concatenating Hl's various inputs into a single tensor simplifies its use.

C. DenseBlocks

When the dimensions of the feature maps vary, it is not possible to utilize the concatenation method. To achieve faster calculation rates, however, CNNs rely heavily on down-sampling of layers, which decreases the size of feature-maps via dimensionality reduction.

DenseNets are partitioned into DenseBlocks to facilitate this, with the feature map dimensions being the same inside each block but the number of filters being varied between them. Transition Layers, which are located between individual blocks, are responsible for halving the total number of channels.

The following equation defines Hl as a composite function that performs three operations in sequence for each layer: batch normalization (BN), a rectified linear unit (ReLU), and a convolution (Conv).

D. Growth Rate

The characteristics may be viewed as the network's overall condition. After traversing each thick layer, the feature map expands in size because each layer adds 'K' features to the current global state. This value of 'K' controls how much new data is added to each successive layer of the network and is hence the network's growth rate. The lth layer has k features if each function H l generates a feature map.$\mathbf{k_l = k_0 + k * (l-1)}$ input feature-maps, where k_0 is the number of channels in the input layer. Unlike existing network architectures, DenseNets can have very narrow layers.

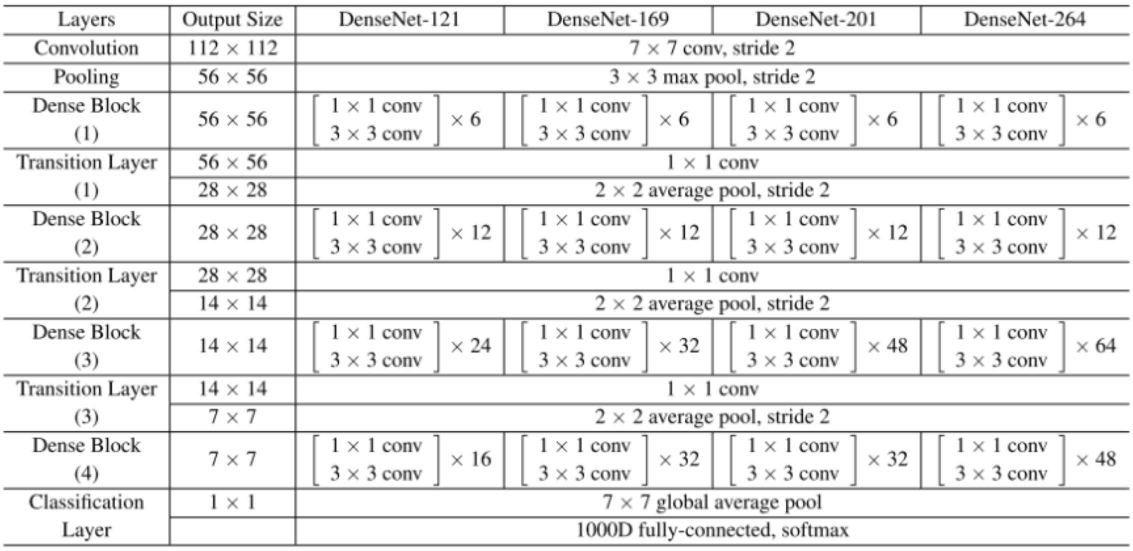
E. Bottleneck Layers

Even though there are only k feature-maps produced by each layer, the number of inputs might be very large. Therefore, before each 3x3 convolution, a 1x1 convolution layer may be added as a bottleneck layer to increase computing efficiency and speed.

F. MaskRepresentation:

An object's spatial characteristics are encoded in its mask. In contrast to the classification and bounding box regression layers, the fully connected layer requires perfect pixel-to-pixel correspondence in order to improve its output. When predicting the mask, Mask R-CNN employs a fully linked network. This ConvNet receives a Region of Interest (RoI) and produces a representation of m*m masks. We also use 1*1 convolution to downscale the mask's channels to 256 so that it may be used for inference on the input image. To feed this mask-predicting neural network with data, we use. The Point of RoI.

Before an area proposal network can "align," it must construct a collection of feature maps with varied sizes and then combine them into a single, consistent one.

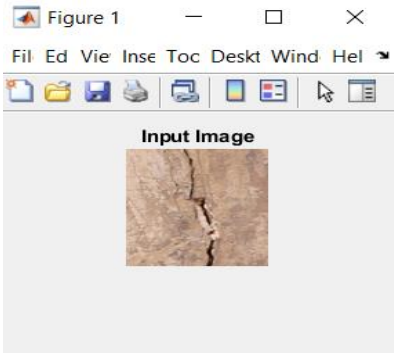


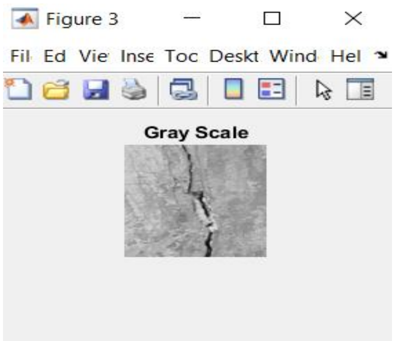
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

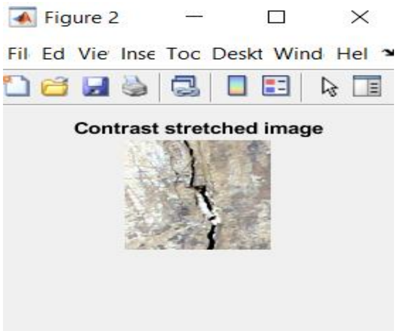
Figure 5: DenseNet-121 Architecture

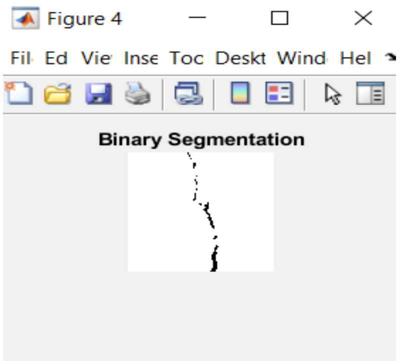
The above table provides a summary of the much architecture used for the ImageNet database. The input matrix is shifted by a certain number of pixels, denoted by "stride." When the stride is set to 'n' (the default is '1'), the filters are shifted by that many pixels.

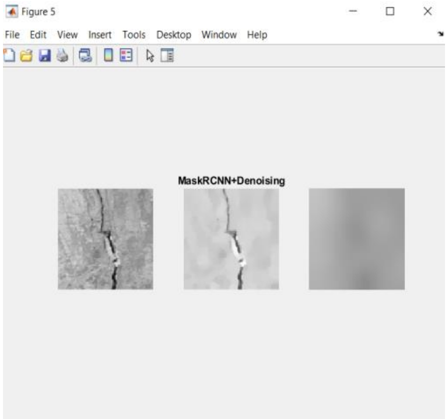
IV. RESULTS AND DISCUSSION

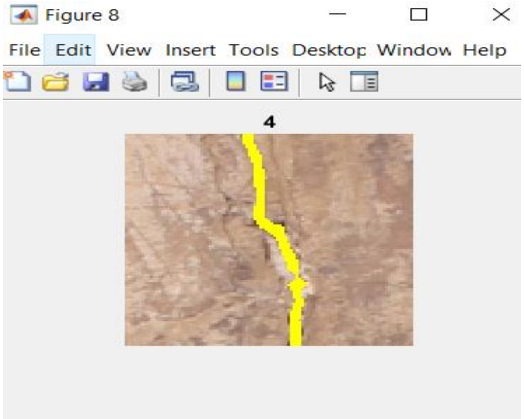




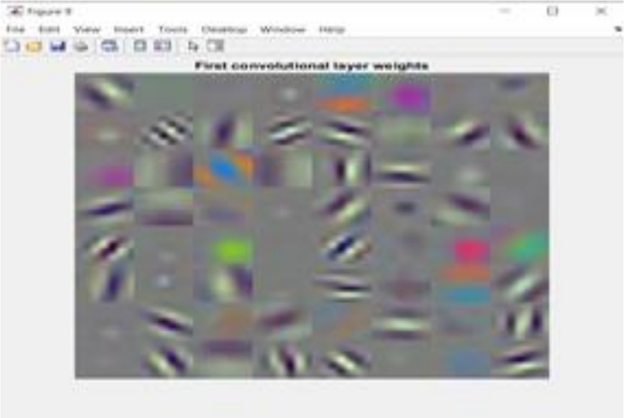




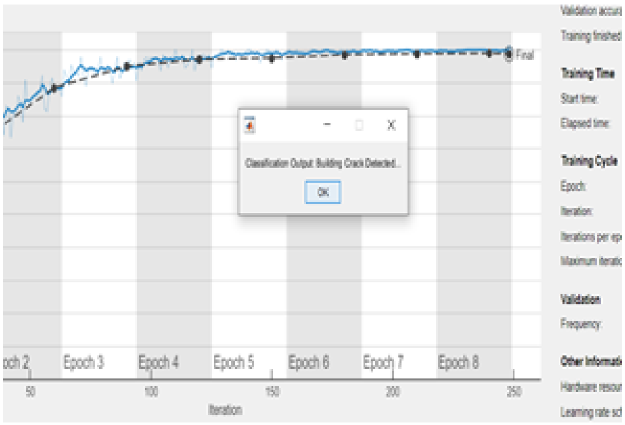




Layer Mapping:


File Edit View Insert Tools Desktop Window Help
First convolutional layer weights

Evaluation Results:


Classification Output: Building Crack Detected...

V. CONCLUSION

Numerous computer-vision use cases focus on cracks because they are common line topologies. Many real-world cracks, such as those in pavement, exhibit poor continuity and low contrast, posing significant difficulties for low-level feature-based image-based crack recognition. In this study, an automated MATLAB-based fracture detecting approach employing Mask RCNN DensNet was developed. The proposed method has a greater rate of accuracy and precision than previously used methods.

VI. REFERENCES

[1] D. Xu, G. Xu and G. Xu, "Crack Identification Algorithm Based on MASK Dodging Principle and Deep Learning," *2019 Chinese Automation Congress (CAC)*, Hangzhou, China, 2019, pp. 5770-5776, doi: 10.1109/CAC48633.2019.8996715.

[2] W. A. Altabey, S. A. Kouritem, M. I. Abouheaf and N. Nahas, "Research in Image Processing for Pipeline Crack Detection Applications," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, 2022, pp. 1-6, doi: 10.1109/ICECCME55909.2022.9988417.

[3] S. Ogawa, K. Matsushima and O. Takahashi, "Crack Detection Based on Gaussian Mixture Model using Image Filtering," *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, Ho Chi Minh City, Vietnam, 2019, pp. 79-84, doi: 10.1109/ISEE2.2019.8921060.

[4] S. B. Ali, R. Wate, S. Kujur, A. Singh and S. Kumar, "Wall Crack Detection Using Transfer Learning-based CNN Models," *2020 IEEE 17th India Council International Conference (INDICON)*, New Delhi, India, 2020, pp. 1-7, doi: 10.1109/INDICON49873.2020.9342392.

[5] P. Kumar, S. Batchu, N. Swamy S. and S. R. Kota, "Real-Time Concrete Damage Detection Using Deep Learning for High Rise Structures," in *IEEE Access*, vol. 9, pp. 112312-112331, 2021, doi: 10.1109/ACCESS.2021.3102647.

[6] G. Li, J. Wan, S. He, Q. Liu and B. Ma, "Semi-Supervised Semantic Segmentation Using Adversarial Learning for Pavement Crack Detection," in *IEEE Access*, vol. 8, pp. 51446-51459, 2020, doi: 10.1109/ACCESS.2020.2980086.

[7] J. M. Z. Maningo *et al.*, "Crack Detection With 2D Wall Mapping For Building Safety Inspection," *2020 IEEE REGION 10 CONFERENCE (TENCON)*, Osaka, Japan, 2020, pp. 702-707, doi: 10.1109/TENCON50793.2020.9293727.

[8] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang and S. Wang, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection," in *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498-1512, March 2019, doi: 10.1109/TIP.2018.2878966.

[9] Y. Yu, H. Guan, D. Li, Y. Zhang, S. Jin and C. Yu, "CCapFPN: A Context-Augmented Capsule Feature Pyramid Network for Pavement Crack Detection," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3324-3335, April 2022, doi: 10.1109/TITS.2020.3035663.

[10] H. Yao, Y. Liu, X. Li, Z. You, Y. Feng and W. Lu, "A Detection Method for Pavement Cracks Combining Object Detection and Attention Mechanism," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22179-22189, Nov. 2022, doi: 10.1109/TITS.2022.3177210.

[11] E. Buza, A. Akagic and I. Besic, "Image-Based Crack Detection Using Sub-image Technique," *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, Turkey, 2019, pp. 614-618, doi: 10.23919/ELECO47770.2019.8990615.

[12] S. -M. Kang, C. -J. Chun, S. -B. Shim, S. -K. Ryu and J. -D. Baek, "Real Time Image Processing System for Detecting Infrastructure Damage: Crack," *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2019, pp. 1-3, doi: 10.1109/ICCE.2019.8661830.

[13] J. J. Dhule, N. B. Dhurpate, S. S. Gonge and G. M. Kandalkar, "Edge Detection Technique Used for Identification of Cracks on Vertical Walls of The Building," *2015 International Conference on Computing and Network Communications (CoCoNet)*, Trivandrum, India, 2015, pp. 263-268, doi: 10.1109/CoCoNet.2015.7411196.

[14] Prasetyo, E. M. Yuniarto, P. Suprobo and A. Tambusay, "Application of Edge Detection Technique for Concrete Surface Crack Detection," *2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia, 2022, pp. 209-213, doi: 10.1109/ISITIA56226.2022.9855280.

[15] Villanueva *et al.*, "Crack Detection and Classification for Reinforced Concrete Structures using Deep Learning," *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022, pp. 1-6, doi: 10.1109/CONIT55038.2022.9848129.