IMAGE-BASED STRUCTURAL DAMAGE RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Jiawei ZHANG[1], Reachsak LY[2], Weijian ZHAO[3], Yunyi LIU[4], Jiangpeng SHU[5]

1. Research Assistant, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
2. Research Assistant, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
3. Professor, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
4. Professor, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
5. Professor, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China

Corresponding author email: jpeshu@zju.edu.cn

Abstract

Crack detection and classification is an important task in monitoring and inspection of civil engineering structures. The spatial characteristics and types of cracks are significant indicators to assess and evaluate the health of existing buildings and infrastructure. In recent years, several different methodologies of crack inspections and classification were proposed. However, the studies in the field of crack damage types are still limited and there is also room for improvement for accuracy of the classification. This paper proposed a more effective image-based structural damage recognition system with three baseline recognition tasks: spalling condition check, component types identification and damage types determination. A total of 22900 images are selected from the PHI Challenge 2018 by Pacific Earthquake Engineering Research (PEER) Task 1 and Task2 competition and are manually labeled into the three recognition tasks respectively. The convolutional neural networks and Transfer Learning-based Resnet34 is introduced and applied in this recognition task. The trained CNN classifier is also been integrated into the cloud and enables the system into a web-based crack classification platform. The model achieved 90% accuracy for the multiclass classification task of Damage Types determination and 97% in Component types identification task. For binary class classification task like the damage check and spalling condition check and types of damages, the model manage to achieve over 90 % of accuracy rate. The results illustrate that the proposed method shows high accuracy and robust performance and can indeed detect and classify damage types efficiently.

Keywords: *CNNs, Crack detection, Classification, web-based, Transfer Learning.*

1. Introductions

Numerous Civil infrastructures have gradually approached their design life expectancy which could cause harmful effects to the safety of surrounding people. Therefore it is necessary to effectively check and evaluate the integrity of the civil structure. In the past, different manual inspections methods were introduced but there are problems with unreliable inspection results as well as the time it takes of the inspections. Moreover, the inspector must get close to the structure for better inspection which could cause safety concerns if the structure is under serious damage. Thus, due to the increase of labor costs and low efficiency of manual inspection, being able to continuously and automatically monitor the structure with the minimum amount of manpower has become a crucial research direction in recent years.(Abdel-Qader et al. 2006; Abudayyeh, Al Bataineh and Abdel-Qader 2004; Yang, Yang and Huang 2015)

In the last decades, researchers and engineers in the field of civil engineering have noticed the promising prospects and innovative technological potential of deep learning-based approaches.(DeVries, Viégas, Wattenberg and Meade 2018; Spencer, Hoskere and Narazaki 2019) .Structural damage inspection is essential for the safety of in-service civil structures, and thus many research groups have utilized the deep learning-based approaches to conduct damage detection on a variety of structures.

One of the important applications of deep learning in Civil Engineering is the task of structural health monitoring is the assessment of types and conditions of cracks. Crack classification is an approach to find the specific crack type using machine learning algorithms. Compare to Crack detection which identifies or recognizes the presence of a crack, crack classification classifies the crack based on the feature extracted from the crack region. The renowned supervised learning algorithms applied for crack classification are Support Vector Machine (SVM) (Thai, Hai and Thuy 2012) K Nearest Neighbors algorithm (KNN) (Taneja et al. 2014) Extreme Learning Machine (ELM) (Huang, Zhu and Siew 2006) Gao et al. developed a crack classification system with four baseline recognition tasks: component type identification, spalling condition check, damage level evaluation, and damage type determination. To prevent overfitting, Transfer Learning (TL) based on VGGNet (Visual Geometry Group) and applied using two different strategies, namely feature extractor and fine-tuning. (Gao & Mosalam, 2018)

This paper purpose a new image-based structural damage recognition model using Deep Convolutional Neural Networks with more efficient classification accuracy with more variety of damages types classes. In this article, we focus on three main damages recognition tasks as follows, spalling condition check, component types identification, and damage types determination. (Figure 1)

2. Methodology

2.1 Proposed Method

The classification model is designed to perform the following tasks: (1) binary classification task for spalling condition check, (2) binary classification task for component type identification, (3) multiclass classification task for damage type determination and to be trained by a modified Transfer Learning-based architecture, ResNet34 with 22900 images data.

2.2. Spalling Condition

Spalling condition is a binary classification task with two classes: spalling and no spalling. The spalling condition pertains to the loss of cover of the component's surface. (Figure 2)

2.3 Damage Types Determination

This paper classified damage types into 7 different classes: none, Flexural (bending) cracks ， shear cracks, shrinkage crack, settlement crack, Corrosion crack, and Alkali aggregate crack.
The model will first check if the images consist of any damages before continue to determine the types of cracks represent in the image if any. (Figure 3)

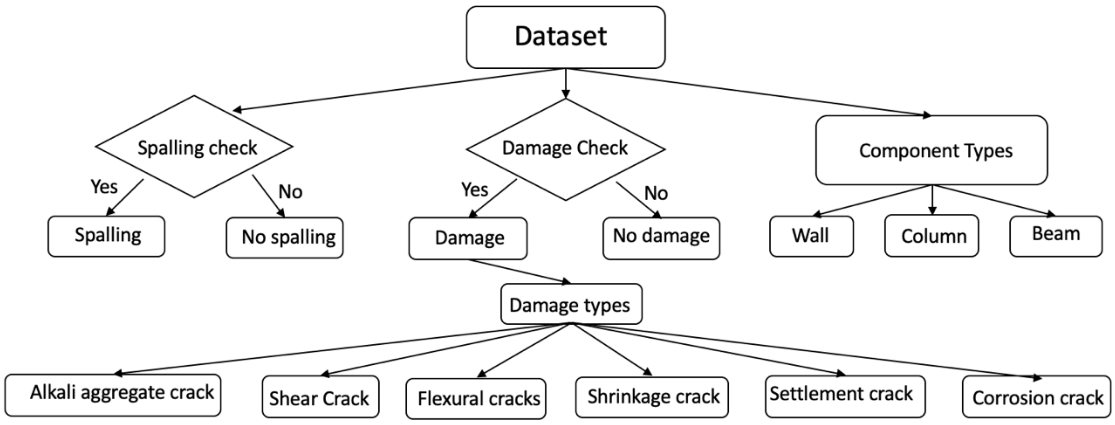

Figure 1.Hierarchy tree of the classification task by the proposed model

2.4 Component Type identification

Component type identification is a binary classification task with three classes: beam column and wall. (Figure 4)

2.5 Databank

A total of 22900 images were collected from data provided by the 2018 PHI Challenge in the Task 1: scene level identification and Task 2: Damage State Check with the image resolution of 224x224.Among them, there are 3600 spalling images, 3800 non-spalling images, 1600 images of walls, 2000 images of beams, 1900 images of column, 4000 concrete crack images, 4000 non-crack images, 500 shear crack images, 300 flexural crack images, 300 shrinkage crack images, 300 settlement crack images, 300 Alkali aggregate crack and 300 corrosion crack images. The proportion of the training set and testing set is 4:1.

To ensure a better generalization ability of the trained model, the crack image dataset is carefully selected with a variety of cracking conditions. Each image is needed to manually annotate into 10 different classes for achieving three recognition tasks: spalling condition check, component types identification, and damage types determination.

Figure 2.Spalling Condition

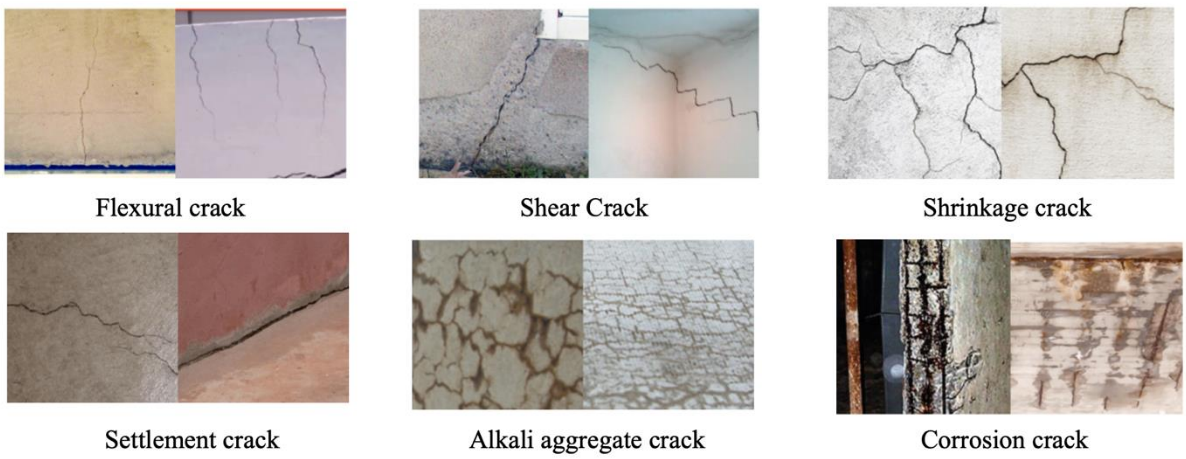

Figure 3.Damage Types

Column Wall Beam

Figure 4.Component Types

3. Network Architecture

3.1 Deep Residual Learning

Instead of hoping every few stacked layers directly fit a desired underlying mapping, He et al. explicitly let these layers fit a residual mapping. The formulation of $\mathcal{F}(x) + x$ can be realized by feedforward neural networks with shortcut connections. Shortcut connections are those skipping one or more layers shown in Figure 5.The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers. (He et al. 2016)

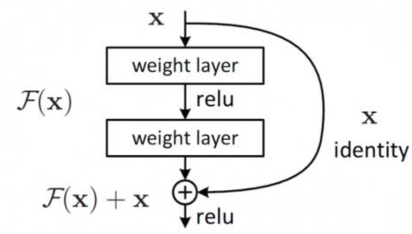

Figure 5.Residual learning: a building block

The Skip Connections between layers add the outputs from previous layers to the outputs of stacked layers. This results in the ability to train much deeper networks than was previously possible. By implementing this method the amount of training time decrease, and the model has also a better convergence performance resulting in a more improved accurate final prediction in the final layer. (He et al. 2016) test their ResNet architecture network with 100 and 1,000 layers on the CIFAR-10 dataset. The model is also tested on the ImageNet dataset with 152 layers, which still has fewer parameters than the VGG network, one of the very popular Deep CNN architecture.

4. Transfer Learning

4.1 Transfer Learning and related terminology

Transfer learning enable researcher to utilize knowledge from previously learned tasks and apply them to newer, related ones. If we have significantly more data for task T1, we may utilize its learning, and generalize this knowledge (features, weights) for task T2 (which has significantly less data).In case for computer vision related task, as usual the first several layers of the convolutional neural network are more generalized like detecting certain low-level features, such as edges, shapes, corners and intensity, that can be shared across tasks, and thus enable knowledge transfer among tasks. And the features extracted from the latter layers are more relevant to the target recognition task.

AdaptiveAvgPool2D: Applies a 2D adaptive average pooling over an input signal composed of several input planes. The number of output features is equal to the number of input planes.

AdaptiveMaxPool2D: Applies a 2D adaptive max pooling over an input signal composed of several input planes. The number of output feature is equal to the number of input planes.

BatchNorm1D: Applies Batch Normalization over a 2D or 3D input (a mini-batch of 1D inputs with optional additional channel dimension) (Ioffe and Szegedy 2015).

Dropout: During training, randomly zeroes some of the elements of the input tensors with probability p using a sample from a Bernoulli distribution. Each channel will be zeroed out independently on every forward call (Hinton et al. 2012).

To utilized the model and adapt to this proposed damages recognition task, the parameters in the former two blocks of ResNet34 (Figure 6) were frozen and the parameters in the latter two blocks were retrained before adding new-trained parameters at the end of the model. (Figure 7) The parameters of the two blocks maintain the extraction of their image features, but only the parameters of the subsequent blocks are fine-tuned to extract the graphic features of the cracks. Also, to enhance the generalization ability of our model, we first removed the previously fully connected layer and connected the average pooling layer and the maximum pooling layer to extract the learned average and maximum response crack graphic features and making a join for the integration of features. The output is then connected to the BatchNorm1d layer, which regulates the feature distribution of the input to promote gradient propagation. To enhance the model's generalization ability for different data, we also connected the Dropout layer to randomly blind some neurons to avoid overfitting of the current data set (equivalent to a regularization operation). Finally, the original fully connected layer is connected again, and the number of types predicted by the output is changed to the number of types that need to be distinguished for our crack identification.

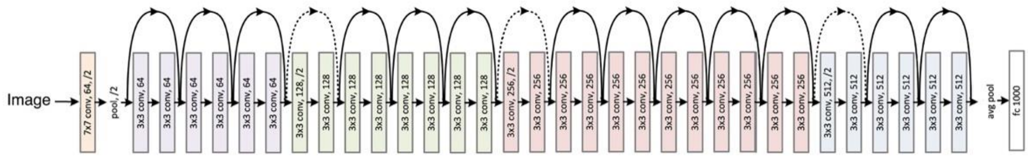

Figure 6.Original ResNet34 architecture

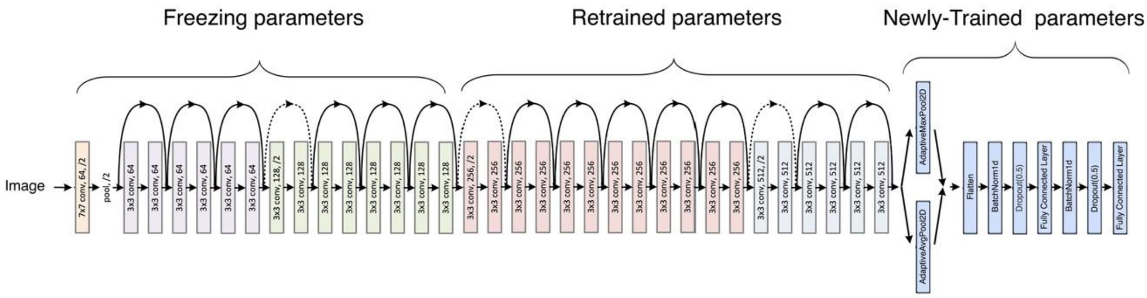

Figure 7. Proposed architecture

4.2 Model Training

This section describes the training process of the ResNet-34 Model including, optimization methods, hardware configuration, etc. All the tasks described in this article are performed on a workstation (CPU: Intel Xeon E5-2678 v3@2.5GHz RAM 64G, GPU: NVIDIA RTX2080TI-11G).

The CNN architecture used in this article is ResNet 34, a variant of ResNet architecture with 34 layers. The model used is initialized by RetNet34 that is pre-trained on the ImageNet1000 class image dataset. The model trained for 50 epochs with CrossEntropyLoss mainly used as the Criterion. This paper uses the Adam algorithm to optimize the model. The Adam algorithm is a combination of the Momentum algorithm and the RMSprop algorithm. Weight Decay are assigned by 5e-5 with the learning rate of 1e-4, betas= (0.5, 0.999).

To prevent overfitting and improving the model performance, Data Augmentation was also implemented by creating transformed versions of images in the training dataset that belong to the same class as the original image, including a range of random operations from the field of image manipulation, such as shifts, reflect, flips, zooms and ColorJitter which randomly changes the brightness, contrast, saturation, and hue of the images. (Figure 8)

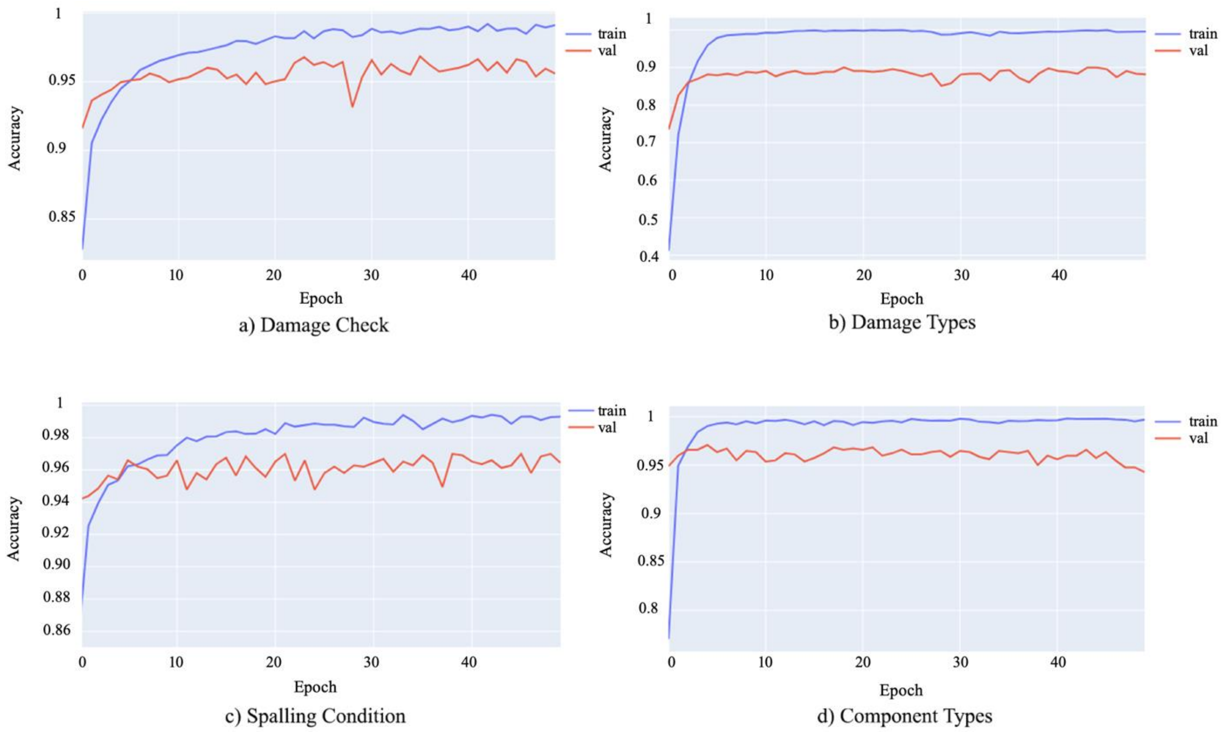

Figure 8.Accuracy history of the four recognition task

5. Experiments and Result

This classification model is evaluated by the Accuracy, Precision, Recall and F1 score. Accuracy is defined by the number of correct predictions divided by the number of the prediction made. In this experiment, Component Types identification and Damages type determination tasks are multiclass classification tasks while spalling condition check, damage check task are binary classification tasks. And according to the label data and the predicted data true positive, false positive, true negative, false negative can be obtained and Recall, Precision, and F1 score can be computed by the following equation:

$$R = \frac{TP}{TP+FN} \tag{1}$$

$$P = \frac{TP}{TP+FP} \tag{2}$$

$$F1 = \frac{2 \times P \times R}{P+R} \tag{3}$$

From the experiment result as shown in Figure 9, even with the complex environment and variety of the background of the image, the model still has achieved a quite good accuracy in Damage check and spalling condition check task with 96.84% and 96.99% accordingly.

However during the component identification task training, we noticed that the training accuracy has reached nearly 100% accuracy in the first few epochs which are the sign of overfitting, and the testing accuracy of the whole network is relatively low. One of the many possible reasons might be the nature of the training set and test set are far too different which causes the inconsistency between feature distribution of the training set and test set. The other reason maybe because of the noise data in the sample is too interfered so that the model remembers the noise features, rather than learning the feature. Besides, in the component identification task, we can also notice that the recall score of beams is lower than the walls and columns counterpart. This is because beams and columns are strip objects in the picture. If a horizontal beam is photographed vertically, it is easy to mistake it as a column, and vice versa. This also made a significant impact on the final accuracy score and directly affects the accuracy of a single output. In the future, we will make a data set with multiple outputs for training to improve recognition accuracy.

In terms of damage types determination task, the model achieved an overall accuracy of 90%. This due to the difficulty in identifying the kind of damage that is too similar. For instance, alkali-aggregate cracks and shrinkage cracks are all crazing-type crack and flexural bending cracks and shear cracks are sometimes difficult to distinguish. If the model only ask to detect between, flexural bending, alkali-aggregate and corrosion damage, which can be distinguished clearly, the accuracy will be greatly improved.

Task	Classification Type	Precision	Recall	F1	Accuracy
Damage Check	Binary	0.9588	0.9615	0.9602	0.9684
Spalling Condition Check	Binary	0.9616	0.9677	0.9647	0.9699
Component Types identification	Multiclass	-	-	-	0.9708
Wall	Binary	0.9782	0.9711	0.9746	0.9829
Beam	Binary	0.9741	0.9617	0.9679	0.9817
Column	Binary	0.9618	0.9773	0.9695	0.9769
Damage types determination	Multiclass	-	-	-	0.9000
Shear Crack	Binary	0.9132	0.9524	0.9324	0.9093
Flexural Crack	Binary	0.9216	0.7344	0.8174	0.9512
Shrinkage Crack	Binary	0.9063	0.8788	0.8923	0.9837
Settlement Crack	Binary	0.7778	0.8077	0.7925	0.9488
Alkali aggregate crack	Binary	0.9512	0.9512	0.9512	0.9907
Corrosion crack	Binary	0.9091	1.0000	0.9524	0.9930

Figure 9.Accuracy history for recognition tasks

6. Integrating the Trained CNN into the Web-based classification platform

By using a python web framework module, Flask, We have integrated the trained model to the cloud. This resulted in a cloud-based platform that allows users to detect cracks in real-time where internet connection available on both computers and mobile devices.(Figure 10)

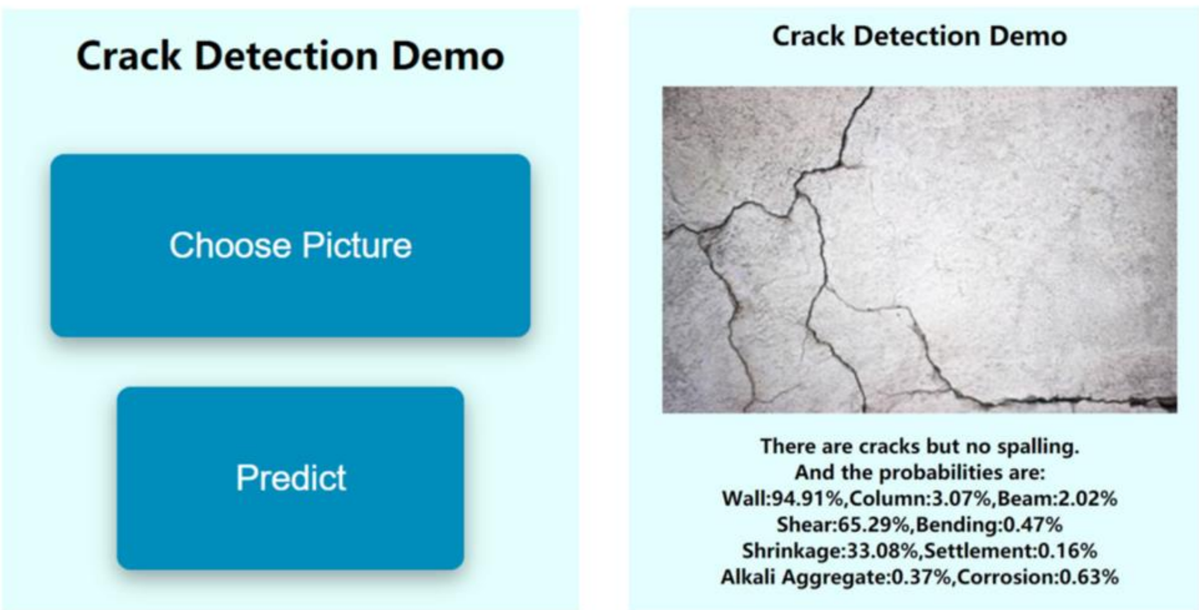

Figure 10.Interface of the Web-based crack detection demo

7. Conclusion

In this article, we first briefly discussed the importance of utilizing Deep Learning knowledge in civil engineering applications, namely, Structural Health Monitoring. Then, we make a review of a well-known deep CNN architecture, namely ResNet and introduced the concepts of Transfer Learning. We introduce a system with a three-baseline damage recognition task, spalling condition check, component types identification, and damage types determination. In order to achieve a good recognition performance toward small data set tasks, instead of training deep CNN from scratch, deep Transfer Learning with ResNet 34 pre-trained model is implemented.

The model achieved high accuracy in both multiclass classification tasks (Damage Types determination and in Component types identification task) and binary class classification tasks like the damage check and spalling condition check and types of damages the model manage to achieve over 90 % of accuracy rate respectively. The trained CNNs have also been integrated into the web which results in a web-based crack classification system.

In the future, we aim to work on the improvement of the accuracy of the multiclass classification task and integrate the trained CNNs with the Unmanned Aerial Vehicle for real-time crack detection.

Reference

Abdel-Qader, I., Pashaie-Rad, S., Abudayyeh, O., & Yehia, S. (2006). PCA-Based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*. https://doi.org/10.1016/j.advengsoft.2006.06.002

Abudayyeh, O., Al Bataineh, M., & Abdel-Qader, I. (2004). An imaging data model for concrete bridge inspection. *Advances in Engineering Software*. https://doi.org/10.1016/j.advengsoft.2004.06.010

DeVries, P. M. R., Viégas, F., Wattenberg, M., & Meade, B. J. (2018). Deep learning of aftershock patterns following large earthquakes. In *Nature* , 560(7720), 632-634 https://doi.org/10.1038/s41586-018-0438-y

Gao, Y., & Mosalam, K. M. (2018). Deep Transfer Learning for Image-Based Structural Damage Recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768. https://doi.org/10.1111/mice.12363

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.90

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. 1–18. http://arxiv.org/abs/1207.0580

Huang, G. Bin, Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2005.12.126

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*.

Spencer, B. F., Hoskere, V., & Narazaki, Y. (2019). Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. In *Engineering*. https://doi.org/10.1016/j.eng.2018.11.030

Taneja, S., Gupta, C., Goyal, K., & Gureja, D. (2014). An enhanced K-nearest neighbor algorithm using information gain and clustering. *International Conference on Advanced Computing and Communication Technologies, ACCT*. https://doi.org/10.1109/ACCT.2014.22

Thai, L. H., Hai, T. S., & Thuy, N. T. (2012). Image Classification using Support Vector Machine and Artificial Neural Network. *International Journal of Information Technology and Computer Science*. https://doi.org/10.5815/ijitcs.2012.05.05

Yang, Y. Sen, Yang, C. M., & Huang, C. W. (2015). Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis. *Advances in Engineering Software*. https://doi.org/10.1016/j.advengsoft.2015.02.005