Proceedings of the 4th ITB Graduate School Conference
Innovation and Discovery for Sustainability
July 6, 2023

Convolutional Neural Network and Interpretable Deep Learning for Concrete Crack Image Classification

Nayottama Putra Suherman[*], Pramudita Satria Palar & Lavi Rizki Zuhal

Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung,
Jalan Ganesa 10, Bandung 40132, Indonesia
[*]Email: nayotherman@gmail.com

Abstract. The development of unmanned aerial vehicles (UAVs) makes automation of visual tasks possible, such as crack detection. Crack detection has many challenges and, in this work, the utilization of image processing through deep learning-based computer vision is conducted for concrete surface image classification. The widely used deep learning architecture for computer vision is the convolutional neural network (CNN). This paper discusses the creation of CNN models for concrete crack image classification and the role of interpretable deep learning in the model's evaluation. Three convolutional architectures combined with a proposed classifier architecture were trained and evaluated quantitatively and qualitatively. The quantitative evaluation metrics are precision, recall, F1-score, and accuracy. The qualitative evaluation concerns the feature highlight of the model using SmoothGrad. The result is that even though the model with InceptionV3 has the best quantitative classification metric values (accuracy of 96%), the one with EfficientNetV2S has the best feature highlight. Thus, the model considered the best is the one with EfficientNetV2S since the accuracy is already considered high (94%). This highlights the importance of qualitative evaluation on a deep learning-based computer vision model to ensure the correct feature is considered as the deciding point of classification.

Keywords: *convolutional neural network; interpretable deep learning; concrete crack.*

1 Introduction

In aerospace engineering, there are numerous possibilities for innovation to solve or automate problems in various fields. One of the developing industrial applications is the utilization of unmanned aerial vehicles (UAVs) for visual tasks. This innovation is convenient because UAVs can do some mundane work that might be high risk, such as inspecting tall buildings for cracks. Another example is inspecting a very long pipeline visually for certain signs of structural health, which might take weeks if done by humans. Crack detection itself possesses several challenges and has so many layers of problems that need to be solved. The discussion of crack detection-related problems is discussed by Yao *et al.* in [1] and specifically using image processing is discussed by Mohan and Poobal in [2]. These tasks are possible to do just by putting cameras or other sensors on a UAV for a safer and automated maintenance operation. However, putting a camera on the UAVs is not the only aspect to engineer this automation.

To do this, the development of the computer vision methodology to detect the necessary visual features of interests needed to be developed. The future of computer vision is through the deep learning approach to solve visual tasks. The main goal of computer vision is to automate the said visual tasks which might perform much better than what humans could do. There are several ways to utilize the deep learning concept for computer vision but the most widely used one is the Convolutional Neural Network (CNN). The main idea of using CNN is to try and detect the important features in the image from the various convolutional layers [3]. However, there is a present weakness in using the deep learning approach, which is the black box problem. The black box problem is the problem stemmed from the complexity of the neural network that we as a user do not know how a neural network model decide an output [4]. When developing a CNN model, we need to be sure that the model highlights the correct features. Thus, in this work, the CNN model is created for concrete surface image classification and with a more comprehensive model evaluation.

2 Problem Statement and Related Works

The concrete surface image classification is aimed to detect whether there are cracks in the said surface. The main problem is to make a CNN model for classification and investigate whether the model highlights the correct features, which are supposed to be the cracks. This work does not discuss the segmentation of cracks in the images, like using R-CNN, or detection but rather to see which features are the deciding point in the classification. This demonstration hopefully will give light on the importance of a qualitative evaluation, which is the feature highlight investigation of a neural network model for computer vision tasks. This means that this work is not comparable with those that segment the cracks first and then classify the image, such as the work of Billah *et al.* in [5]. This work also only discusses CNN-based computer vision methodology and its utilization in the subject of concrete crack detection so another method such as the work of Wang *et al.* in [6] is not discussed.

There are some related works in utilizing CNN for concrete crack image classification. Su and Wang in [7] provide several comparisons of some well-established CNN architectures and their respective performances and Falaschetti *et al.* in [8] even propose a new architecture, albeit both without investigating the feature highlight of the models. Özgenel and Sorguç in [9] provide a dataset of concrete crack and compares several CNN architectures' classification performance, even though, again, does not evaluate their qualitative performance. Another utilization of CNN, although for segmentation, is the work of Yang *et* *al*. in [10], which used CNN for detecting and labelling the crack in the image. This work creates and compares several CNN models to classify concrete surface images to find cracked surfaces. The model is composed of a well-established convolutional architecture and a proposed classifier architecture. The models are evaluated through a quantitative and a qualitative evaluation to decide which model is the best.

3 Methodology

3.1 Dataset

The dataset is obtained from the aforementioned related works. For the training of the CNN model, the dataset used is only from the dataset used by Özgenel and Sorguç in [9] as the data is publicly available. Some examples of the training dataset are displayed in Figure 1 with the images with no crack labelled ‘Negative’ and the images with a cracked surface labelled ‘Positive’. The dataset for testing the model, which is the dataset that has not been used in the training process at all, is the combination of the dataset by Özgenel and Sorguç in [9] and by Yang *et al.* in [10]. This is to evaluate the model with data outside one dataset to ensure the model does not overfit by introducing a different dataset in its evaluation. The concrete surfaces on the training dataset have several challenges, such as motives or concrete patterns that might be mistaken as a crack.

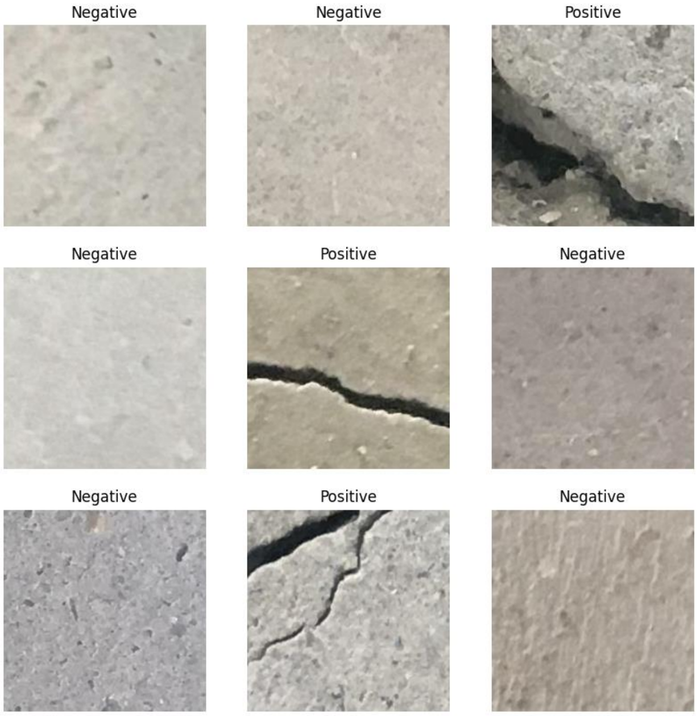

Figure 1 Some image examples from the training dataset for both the cracked and non-cracked concrete surfaces.

The number of images in the training dataset is 16000 images with half of them having a cracked surface and the others not. The number of images in the test dataset is 624 images with half of them having a ‘Positive’ label and the other half not. During the training of the models, the validation dataset is created by separating 20% of the training dataset to assess the validation accuracy of every epoch. The images then are read as RGB images with the size of 256 × 256.

3.2 Convolutional Neural Network Architecture and Hyperparameters

The convolutional neural network architecture consists of two components, the convolutional layers and the classifier layers (which usually are in the form of dense layers). In this work, three established convolutional neural network architectures are compared. Every one of these architectures has its own classifier components which are not used. Instead, the classifier component used is a new proposed classifier architecture, which will be elaborated on later.

The three established convolutional architectures chosen are ResNet50V2 by He *et al.* in [11], InceptionV3 by Szegedy *et al.* in [12], and EfficientNetV2S by Tan and Le in [13]. These three architectures are chosen because their number of parameters sits around 20 million parameters. The output of each convolutional layer is then connected with a proposed classifier architecture. Since one of the goals is to compare the performance of the feature highlight between convolutional architectures, the classifier component architecture of all the models is the same. The architecture of the classifier component is presented in Figure 2. The activation functions of each convolutional layer are as what their default is and the activation functions for the classifier layers are as mentioned in Figure 2. Since the classification is only between two classes, cracked and not cracked concrete surfaces, the activation of the output is set to be a sigmoid function.

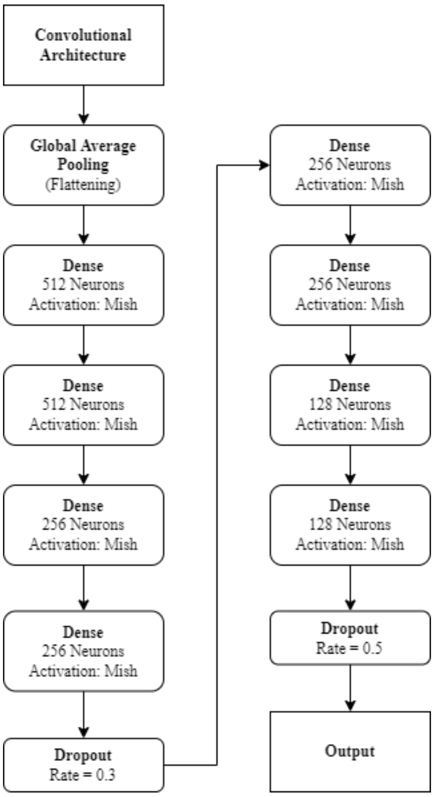
Convolutional Architecture
Global Average Pooling (Flattening)
Dense 512 Neurons Activation: Mish
Dense 512 Neurons Activation: Mish
Dense 256 Neurons Activation: Mish
Dense 256 Neurons Activation: Mish
Dropout Rate = 0.3
Dense 256 Neurons Activation: Mish
Dense 256 Neurons Activation: Mish
Dense 128 Neurons Activation: Mish
Dense 128 Neurons Activation: Mish
Dropout Rate = 0.5
Output

Figure 2 The proposed classifier architecture.

The training of the model is done with early stopping, which means if, in the training process, the validation accuracy does not improve after a set of epochs, the training is stopped. A reducing learning rate mechanism is also applied in the training process, which means if the validation accuracy does not improve after some epochs, the learning rate is reduced. The optimizer used for the training process is the Adaptive Moment Estimation (Adam) algorithm. The cost function used in the training process, since this is a binary classification problem, is binary crossentropy.

3.3 Quantitative and Qualitative Evaluation

The quantitative evaluation consists of four things, which are accuracy, precision, recall, and F1-score. Before the explanation of those metrics, there are some concepts needed to be explained first, which are true, false, positive, and negative. True and false denote whether the prediction is correct or not. Positive and negative denote whether the prediction predicts the image in that certain class. For example, false positive (FP) prediction means that it is falsely predicted as positive, which means the true value is negative, but the model predicts it as positive. Accuracy concerns the correct prediction out of all the predictions as in Eq. (1). Precision is the metric that asses correctly predicted positive data out of all the positive predicted data, which equation is shown in Eq. (2). Recall is the metric that asses correctly predicted positive data out of all actual positive data, which equation is shown in Eq. (3). The F1-score is a metric that combines precision and recall in a single metric because of the precision-recall trade-off, which is as in Eq. (4).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{4}$$

The qualitative evaluation, which is executed by investigating the feature highlight, is done using an algorithm called SmoothGrad proposed by Smilkov *et al.* in [14]. SmoothGrad is the modification of the vanilla saliency map proposed by Simonyan *et al.* in [15] where noise is added to the equation. The basis of SmoothGrad is that it is a sensitivity analysis of the model based on each pixel of the input image to see which pixels in the image contribute the most in classifying to a certain class. An example of the utilization of SmoothGrad to a model is presented in Figure 3. As we can see, the features that the model highlighted are the shell and the head shape. From the evaluation using SmoothGrad, we can say that the model more or less has a good feature highlight performance since it highlights the correct features. If one asks a person whether the picture contains a snail and why, they might answer because of the shell and the head shape.

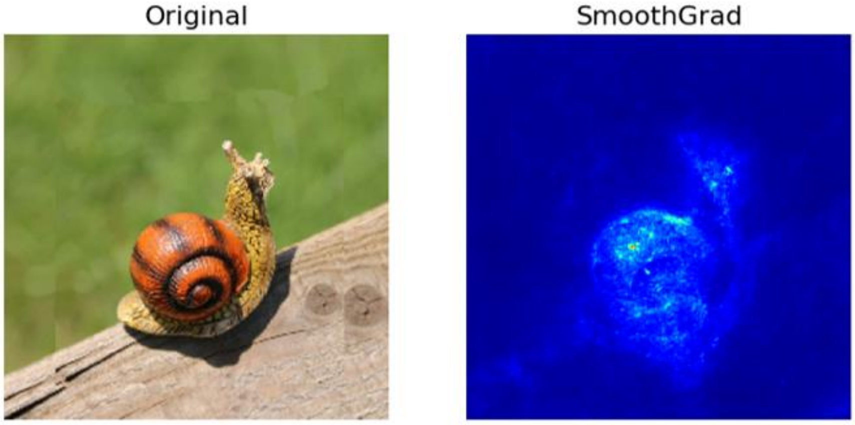

Figure 3 An example of SmoothGrad application on VGG19 snail classification's qualitative evaluation. The SmoothGrad highlights the features that the model deemed important.

4 Result and Analysis

The results of the quantitative evaluation of every CNN model that are compared to the test dataset are presented in Table 1. For the qualitative evaluation of the models, two sets of images, with 4 images in each set, become the input for the model. The result of the feature highlight of each model is displayed in Figure 4 and Figure 5.

Table 1 The quantitative evaluation results of the three CNN models.

Model	Class	Precision	Recall	F1-Score	Accuracy
ResNet50V2	Negative	0.93	0.99	0.96	0.96
	Positive	0.99	0.93	0.96	
InceptionV3	Negative	0.93	1.00	0.96	0.96
	Positive	1.00	0.93	0.96	
EfficientNetV2S	Negative	0.90	1.00	0.95	0.94
	Positive	1.00	0.89	0.94	

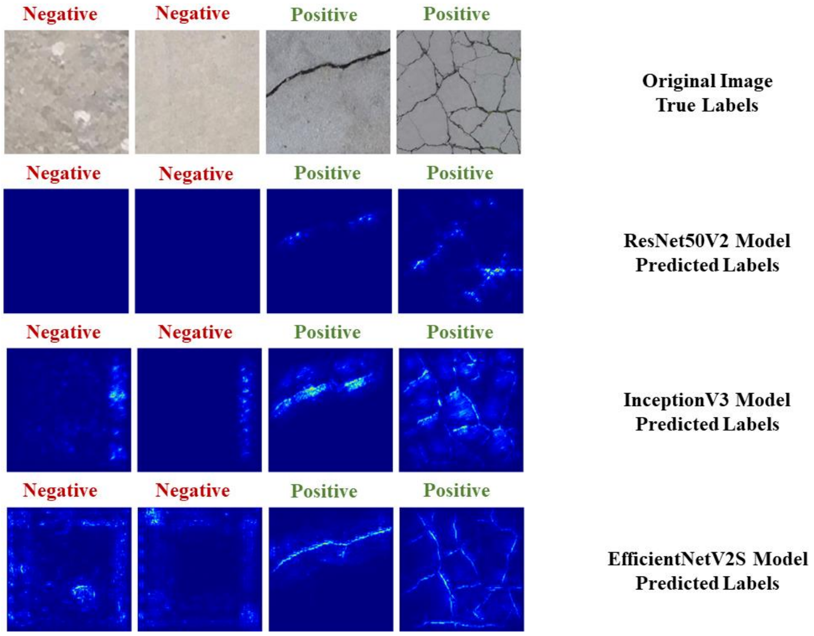

Figure 4 The qualitative evaluation of each model towards image set 1 with each of their predicted labels.

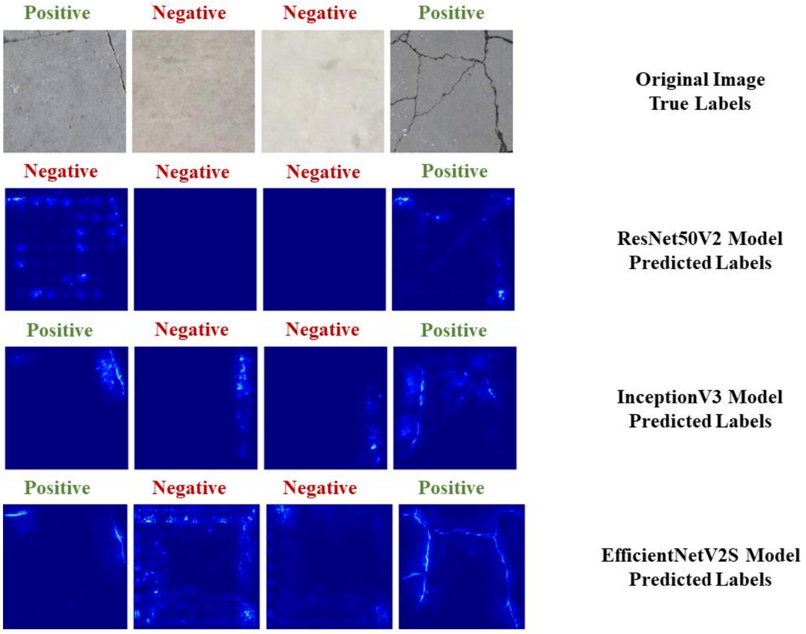

Figure 5 The qualitative evaluation of each model towards image set 2 with each of their predicted labels.

As we can see from the quantitative evaluation results, the model using InceptionV3 convolutional architecture has the best performance in terms of quantitative evaluation. One might also think that the model with ResNet50V2 is a "better" model since it has a higher accuracy than the one with EfficientNetV2S. However, as we see from the model feature highlight in Figure 4 and Figure 5, the model with ResNet50V2 has a poor feature highlight performance. For example, if we look at the rightest image in Figure 4 and Figure 5, we can see that the model with ResNet50V2 does not entirely highlight the cracks. Even InceptionV3 does not entirely highlight the crack on the rightest image in Figure 5. Overall, the model InceptionV3 has a somewhat decent feature highlight, but the concentration of the importance is quite unfocused on the cracks. The best feature highlight is the model with EfficientNetV2S. With all CNN models having decently high accuracies, the best model is EfficientNetV2S considering that highlighting the correct features is an integral part of CNN models.

5 Conclusion

In this work, three CNN models are created for solving the concrete crack image problem. The three CNN models are then evaluated quantitatively and qualitatively using SmoothGrad as the interpretable algorithm. The evaluation resulted in the models using ResNet50V2 and InceptionV3 convolutional architecture having the best classification quantitative metrics values (accuracy of 96%). However, the model with EfficientNetV2S has the best feature highlight performance for the qualitative evaluation. Thus, the model considered the best is EfficientNetV2S since it still has extremely high accuracy (94%) while having the best feature highlight performance. This work also highlights the importance of evaluating the feature highlight aspect of the CNN model since another model might be chosen if the evaluation is not conducted.

References

[1] Yao, Y., Tung, S.-T. E. & Glisic, B., *Crack detection and characterization techniques-An overview*, Struct Control Health Monit, **21**(12), pp. 1387–1413, Dec. 2014, doi: 10.1002/stc.1655.

[2] Mohan, A. & Poobal, S., *Crack detection using image processing: A critical review and analysis*, Alexandria Engineering Journal, **57**(2), pp. 787–798, Jun. 2018, doi: 10.1016/j.aej.2017.01.020.

[3] LeCunn, Y., Bengio, Y. & Hinton, G., *Deep learning*, Nature, **521**, pp. 436-444, May. 2015.

[4] Buhrmester, V., Münch, D. & Arens, M., *Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey*, Nov. 2019.

[5] Billah, U. H., La, H. M. & Tavakkoli, A., *Deep Learning-Based Feature Silencing for Accurate Concrete Crack Detection*, Sensors, **20**(16), p. 4403, Aug. 2020, doi: 10.3390/s20164403.

[6] Wang, B., Zhao, W., Gao, P., Zhang, Y. & Wang, Z., *Crack Damage Detection Method via Multiple Visual Features and Efficient Multi-Task Learning Model*, Sensors, **18**(6), p. 1796, Jun. 2018, doi: 10.3390/s18061796.

[7] Su, C. & Wang, W., *Concrete Cracks Detection Using Convolutional NeuralNetwork Based on Transfer Learning*, Math Probl Eng, vol. 2020, pp. 1–10, Oct. 2020, doi: 10.1155/2020/7240129.

[8] Falaschetti, L., Beccerica, M., Biagetti, G., Crippa, P., Alessandrini, M. & Turchetti, C., *A Lightweight CNN-Based Vision System for Concrete Crack Detection on a Low-Power Embedded Microcontroller Platform*, Procedia Comput Sci, **207**, pp. 3948–3956, 2022, doi: 10.1016/j.procs.2022.09.457.

[9] Özgenel, Ç.F. & Sorguç, A.G., *Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings*, 35th International Symposium on Automation and Robotics in Construction. pp. 693-700 Jul. 2018. doi: 10.22260/ISARC2018/0094.

[10] Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. & Yang, X., *Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network*, Computer-Aided Civil and Infrastructure Engineering, **33**(12), pp. 1090–1109, Dec. 2018, doi: 10.1111/mice.12412.

[11] He, K., Zhang, X., Ren, S. & Sun, J., *Identity Mappings in Deep Residual Networks*, in European Conference on Computer Vision, Leibe, B., Matas, J., Sebe, N. & Welling M. (Eds.), Amsterdam: Springer Cham, Oct. 2016, pp. 630–645. doi: 10.1007/978-3-319-46493-0_38.

[12] Szegedy, C, Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z., *Rethinking the Inception Architecture for Computer Vision*, Dec. 2015.

[13] Tan, M. & Le, Q.V., *EfficientNetV2: Smaller Models and Faster Training*, Apr. 2021.

[14] Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M., *SmoothGrad: removing noise by adding noise*, Jun. 2017.

[15] Simonyan, K., Vedaldi, A. & Zisserman, A., *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, Dec. 2013.