2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

# IMAGE-BASED SEMANTIC RECOGNITION AND SEGMENTATION OF CONCRETE DAMAGES FOR THE ASSESSMENT OF EXISTING CONCRETE STRUCTURES

Fabian Kaufmann[1], Cheng Lu[1], Marius Schellen[1], and Christian Glock[1]
[1]University of Kaiserslautern-Landau, Germany

## Abstract

In this study, an approach to integrate concrete crack and spalling into BIM models using the open data format Industry Foundation Classes (IFC) will be introduced. After a learning-based crack and spalling detection algorithm has been developed, the creation of the BIM model will be demonstrated. The detection algorithm involves a transfer learning approach to minimise the need for annotated training data, while achieving a high prediction accuracy.

## Introduction

Even highly industrialised countries face the problem of an ageing infrastructure in a constantly deteriorating state. Considering road and railway networks, many infrastructure assets include concrete structures. Since the collapse of the Carola Bridge in Germany in 2024, concrete bridges are at the centre of the discussion. It has become obvious that a more in-depth and frequent monitoring will be the only option to prolong the lifetime of our infrastructure. Facing a lack of skilled engineers, increased efficiency in inspection and monitoring will be crucial.

Two core technologies can be considered for more efficient inspection and monitoring. On the one hand, a digital twin based on a BIM model will be beneficial to organize data and information along the lifecycle of infrastructure assets. On the other hand, image-based damage detection will facilitate an automated inspection, and a fluent integration into digital twins.

Cracks and spalling are problematic for concrete structures exposed to the elements. In addition to structures with complex requirements, e.g., in water engineering or in marine environments, bridges are the most common type of structure exposed to the elements. The exposure is combined with high requirements for load bearing capacity and reliability. Hence, this paper will focus on these types of structure, first exploring the state-of-the-art in crack and spalling detection with a dedicated focus on studies integrating defects into BIM models. Furthermore, our method for image-based crack detection will be presented along with a novel dataset. Although more advanced approaches may exist, our approach stands out as it uses a comparatively small number of training images in a fine-tuned U-Net architecture to reliably detect defects. Using a reconstruction procedure, the detected defects can then be converted into BIM objects in the open IFC standard.

## Related work

### Semantic segmentation of damages

In 2018, Gao and Mosalam (2018) used VGGNet-based deep transfer learning for structural damage recognition using feature extractor and fine-tuning to optimise the model showing the potential of transfer learning for image-based structural damage recognition. Similarly, Su and Wang (2020) and Teng et al. (2021) also successfully applied transfer learning to detect concrete cracks. However, their datasets were all limited to close concrete surfaces and lacked complexity.

In the work of Escalona et al. (2019), a U-Net-based neural network architecture is used to detect cracks in pavements. Both works only perform a binary segmentation (crack / no crack), a distinction between crack and spalling has not been investigated. However, the work of Escalona et al. demonstrates approaches that can also be utilised in crack and spalling segmentation on concrete surfaces.

Jierong Cheng et al. (2018) have also developed a crack detection method using U-Net. Although a new cost function based on the distance transform was proposed to increase the accuracy of the prediction, the segmentation accuracy can only reach about 92 %. Li and Zhao (2019) have applied a deep convolutional neural network (CNN) to detect cracks on concrete surfaces and used a comprehensive search strategy to locate the position of the cracks. Nevertheless, the positioning accuracy is not pixel-level due to the size of the sliding window.

Moreover, Li et al. (2019) have utilised transfer learning to initialise the weight and bias parameters of the FCN. The proposed FCN was built by fine-tuning the DenseNet-121 and can be used to detect multiple damages of concrete structures, including cracks, spalling, efflorescence, and holes. Furthermore, Dung and Le Anh (2019) have compared the performance of three different pre-trained CNN models including VGG16, Inception, and ResNet as the encoder of the FCN. The work showed that VGG16 outperforms the other two in crack image segmentation.

Since image-based concrete crack detection requires numerous training images, a benchmark database is essential for this task. Ye et al. (2021) have established a benchmark data set called a bridge crack library (BCL), which contains 11,000 pixel-wise labelled images, including noise images. The BCL was used to train different DNNs for applicability validation. As a result, the BCL was proved to be suitable as a benchmark dataset for the performance evaluation of DNN models.

**BIM for building defects**

Musella et al. (2021) combined BIM and AI technologies to digitise seismic damage in reinforced concrete and brick buildings. They used CNN for image segmentation of structural damage and then developed a new damage-encoding system to import the damage features extracted by AI as parameters into the BIM model and finally evaluated the damage in the form of a table. However, they did not visualise the structural damage in their study and could only represent the severity of the damage in the form of a classification.

Sresakoolchai and Kaewunruen (2021) are the first to integrate BIM and machine learning for railway systems. They utilised axle box acceleration as input data to predict wheel burns, compared the defect location ability under different neural networks, and created a BIM model for a double-track railway project. Finally, they successfully integrated the localisation information from the burn wheel with the BIM model through Dynamo and performed the maintenance function of the 6D BIM model.

Chen et al. (2022) proposed a technical framework that integrates robotics, AI, and BIM for digital twinning of defects, and applied it to a residential building in Hong Kong with good results. They took many photos and videos of the building using unmanned aerial vehicles and then used the SfM module to generate point clouds from them. Building defects were detected using U-Net and finally, the defects were visualised in the BIM model using a newly developed registration algorithm.

Motamedi and N-Yabuki (2017) proposed an IFC data schema to systematically store various types of defects and their relationships with other elements. Although they successfully imported and visualised the damage in the BIM model, they could only manually input the damage information into the IFC schema to create the corresponding damage in the BIM model.

In addition, Artus et al. (2022) developed an automated BIM-based framework for segmentation, modelling, and visualisation of structural damage using the IFC format. After segmenting the damage images using images-based CNN, they used perspective projection equations and the OpenSfm library in Python to create a 3D point cloud model of the damage and finally used coordinate transformation to align the point cloud model of the damage with the existing BIM model. Their models are based on the IFC standard and can be used directly in related subsequent processes.

## Dataset preparation

With the continuous development of deep learning technology, image recognition technology plays an important role in structural health monitoring (SHM) and structural damage assessment. However, the lack of uniform standards for visual attributes and labels and the need for skilled personnel to annotate the data, results in lack of labelled data that can be used for deep learning.

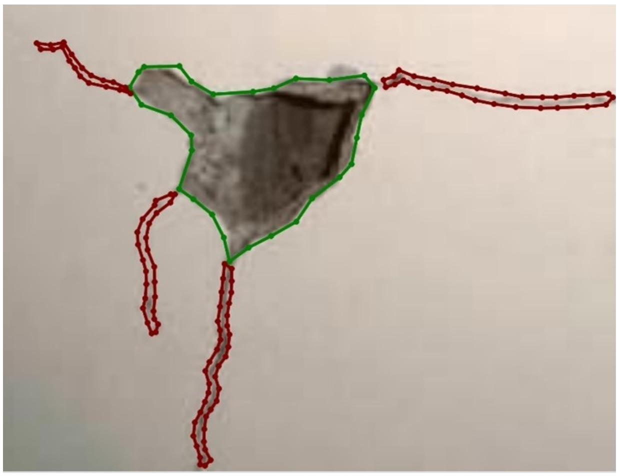

*Figure 1: LabelMe used for image annotation. The red and green polygons are added manually and then exported as the annotation mask for spalling and cracks, respectively.*

Inspired by ImageNet, a team of researchers from the University of California, Berkeley, created the ImageNet database of the Pacific Earthquake Engineering Research Hub (PEER) and made it open source[1]. It contains 36,413 pairs of images taken from different types of structures and damages on various scales along with pixel-level labels (Yuqing Gao and Khalid M. Mosalam, 2020). The dataset used in this study is derived from the damage-type task in the PEER Hub ImageNet database. The dataset contains images of structures with many different damages, e.g. cracks, spalling, or shear damage. Therefore, pixel-level and object-level images containing concrete cracks and spalling are manually selected from the dataset. Structural-level images are not considered in this study.

In this work, the training dataset is labelled using the annotation tool LabelMe. LabelMe is a free open-source graphical tool for creating pixel-level image segmentation annotations[2].

In the LabelMe interface, polygons are used to manually label different structural damages, as shown in Figure 1. The area surrounded by the green polygon is the spalling and the area surrounded by the red polygon is the cracks in the concrete surface. For this study, a total of 2199 images were annotated, of which 2192 images are used for the training and 7 images as the test dataset. All image data are RGB images with 224×224 resolution.

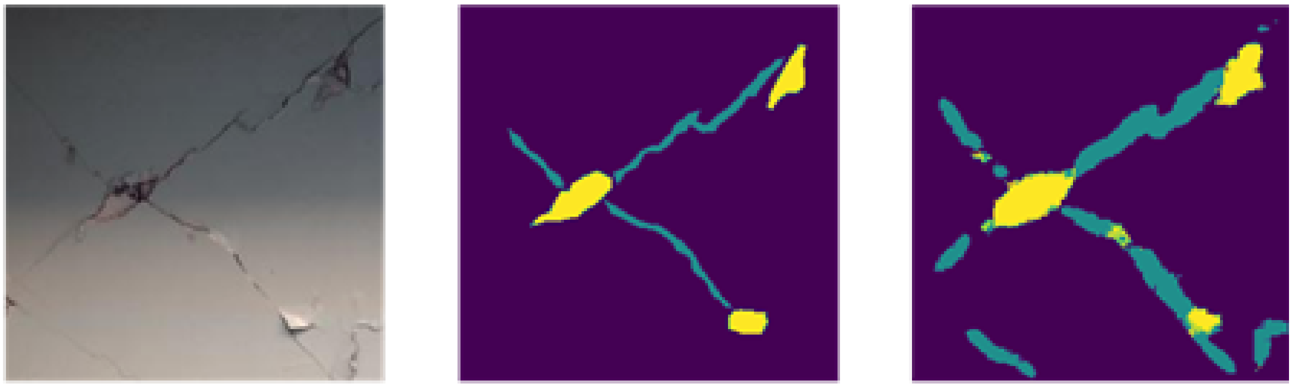

*Figure 2: Training output. Left: original image. Middle: True mask. Right: Predicted mask.*

## A model for semantic segmentation of concrete surface defects

### Dataset preprocessing

Dataset preprocessing includes scaling the images, normalisation, and data augmentation. Bicubic interpolation is used in this work to scale the images to the network's input size. Bicubic interpolation estimates the pixel values of the image on the basis of a weighted average of the surrounding pixels. Image normalisation ensures that the input data has consistent statistical properties, making it easier for the model to learn from the data. It helps in reducing the impact of variations in lighting conditions and colour distributions, making the input data more suitable for accurate and robust machine learning model training and evaluation.

From the wide range of data augmentation techniques, for example, rotation, flip, translation, (Shorten and Khoshgoftaar, 2019), flipping is chosen in this study. It is important to note that excessive augmentation can lead to overfitting, i.e., the model becomes too specialised to the training data and performs poorly on unseen data. Moreover, data augmentation can only achieve a limited improvement in the model's performance since it cannot fully compensate for the scarcity of the dataset. Therefore, only one data augmentation technique was used in this study to ensure that overfitting did not occur.

### Pretrained model for machine learning

Transfer learning is an effective approach to enhance predictive performance and decrease training time when the quantity of available training data is limited (Pan and Yang, 2010), as is the case in this study with only 2199 images available as training data. Instead of training a new model from scratch, a model pre-trained on a large dataset is fine-tuned by adjusting its parameters on a smaller, task-specific dataset. As an image passes through the pre-trained layers, filters are applied to extract features at different spatial scales. After that, the output of the pre-trained layers is then passed through the fine-tuning layers,

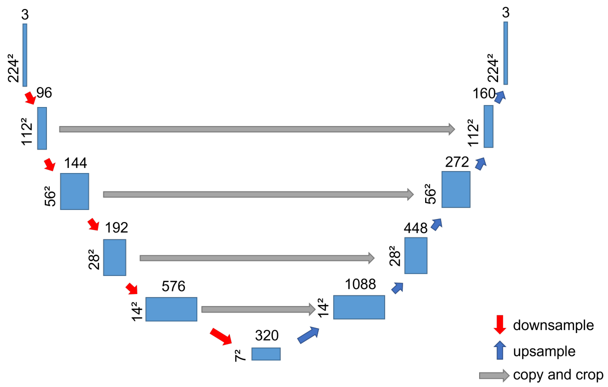


*Figure 3: Illustration of the modified U-Net*

which are adjusted to the target task. The final layer of the model produces the desired result, such as the probabilities of class at the pixel level for semantic segmentation.

In this study, the commonly used MobileNetV2 (Sandler et al., 2018) will be used as the base model. The model was developed by Google researchers in 2018 and was pretrained in the ImageNet dataset (Deng et al., 2009), covering approximately 1.2 million training images and more than 1000 different object categories. The core building blocks of MobileNetV2 (Sandler et al., 2018) are the inverted residual blocks. The inverted residual structure consists of a lightweight bottleneck layer followed by a linear projection and expansion. This design improves the information flow and helps the model capture more complex features efficiently.

U-Net is a popular convolutional neural network (CNN) architecture used for image segmentation tasks in computer vision. The U-Net architecture is based on the fully convolutional network (FCN) and consists of a contracting path (encoder) and an expanding path (decoder), allowing for both localization and segmentation. Instead of using the classical U-Net architecture, the modified U-Net is implemented in this study. The modified U-Net reduces the number of convolutional layers and simplifies the classical U-Net architecture. The contracting path in the modi-

[1] `https://apps.peer.berkeley.edu/phi-net/`
[2] `https://github.com/wkentaro/labelme`

fied U-Net consists of the inverted residual blocks from the pre-trained MobileNetV2. The expanding path keeps the same up-sampling layers in the classical U-Net architecture. Figure 3 illustrates the architecture of the modified U-Net. Each blue box represents a multi-channel feature map. The number of channels is denoted on top of the box. The size is shown on the left of the box.

### Model settings and training

Sparse categorical cross-entropy is implemented as the loss function in this study. It is a commonly used loss function in deep learning for multi-class classification tasks. Sparse categorical cross-entropy is used when the labels are provided as integers instead of one-hot encoded vectors. In other words, the labels are represented as integers from 0 to the number of classes. The loss is minimised when the predicted probability for the true class is high and the predicted probabilities for other classes are low.

Adam is utilised as an optimiser in this study. Adam (Adaptive Moment Estimation) optimiser is an adaptive learning rate optimisation algorithm widely used in training deep learning models. It combines the benefits of both RMSprop and AdaGrad. Adam implements estimations of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network and performs better than the classical stochastic gradient descent procedure (Kingma and Ba, 2014). Adam is known for its fast convergence and efficiency in various deep-learning tasks.

In this study, “Pixel Accuracy” is applied as a metric for evaluating the performance of the model, which is widely used in classification problems. It measures the percentage of correct predictions made by the model out of all the predictions made. The accuracy score is calculated by dividing the number of correctly predicted pixels by the total number of pixels:

$$accuracy = \frac{number\ of\ correctly\ predicted\ pixels}{total\ number\ of\ pixels} \quad (1)$$

Overfitting is a common problem in machine learning, which means that the model is overtrained and fits the training data well, while performing poorly on the test data. Overfitting occurs when the machine learning model is too complex or has been trained too long, while underfitting occurs when the model is too simple to fit the data. Underfitting means that a model is unable to capture the relationships present in the training data. Both Underfitting and Overfitting can lead to inaccurate predictions.

Early stopping is a regularisation technique to prevent overfitting of a model. During training, the model is evaluated on the validation dataset after each epoch. The performance metric used to evaluate the model on the validation dataset is monitored and compared to the best performance seen so far. If the performance on the validation dataset has not improved for a certain number of epochs, then training is stopped, and the weights of the model corresponding to the best performance are used as the final model.

*Table 1: Training and validation accuracy.*

| Dataset division | Training accuracy | Validation accuracy |
|---|---|---|
| 1 | 94.6% | 94.1% |
| 2 | 95.3% | 94.8% |
| 3 | 94.8% | 95.2% |
| 4 | 95.1% | 93.6% |
| 5 | 95.5% | 93.3% |
| 6 | 95.3% | 94.2% |
| Average | 95.1% | 94.2% |

In the study, training will be stopped when the performance on the validation dataset does not improve within 10 epochs.

## Results

### Training and validating results

Figure 2 illustrates an output example generated by the model during the training phase. The input image denotes the image data that was fed into the model for training. The true mask signifies the ground truth labelling of the image data. The predicted mask represents the prediction result made by the model based on the learnt features of the labelled image. From a visual point of view, the model performs well in predicting both cracks and spalling. In addition, it is obvious that the predicted mask not only demonstrates the relatively precise position and shape of the damage within the true mask but also accurately predicts some previously unlabelled damage within the images.

The performance of the model developed in this study has also shown success from a statistical perspective. To this end, the model was trained a total of six times with varying divisions of the training and validation datasets. The accuracy results of the six training sessions are presented in Table 1. The average accuracy of the six training sessions achieves 95.1 % on the training dataset and 94.2 % on the validation dataset, indicating superior performance compared to many studies. Moreover, the accuracy of the model on the training and validation datasets does not appear to be very different, indicating that the model is not overfitted. In particular, all training sessions were completed in 100 epochs, taking less than 10 minutes each.

### Testing results

The high accuracy of a model on the training and validation datasets alone is insufficient evidence to conclude that the model will have a reliable generalisation. Therefore, it is essential to test the model on unseen test datasets to assess its performance and generalisability accurately.

Figure 4 displays seven sets of original images and prediction results from the test dataset. In the analysis of the prediction results, it can be observed that the model demonstrates high recognition and segmentation capability for pixel-level images of concrete cracks in sets 1, 2, 4, and 7. Additionally, the model exhibits satisfactory results for spalling on pixel-level images, except spalling in the lower left part of set 1.

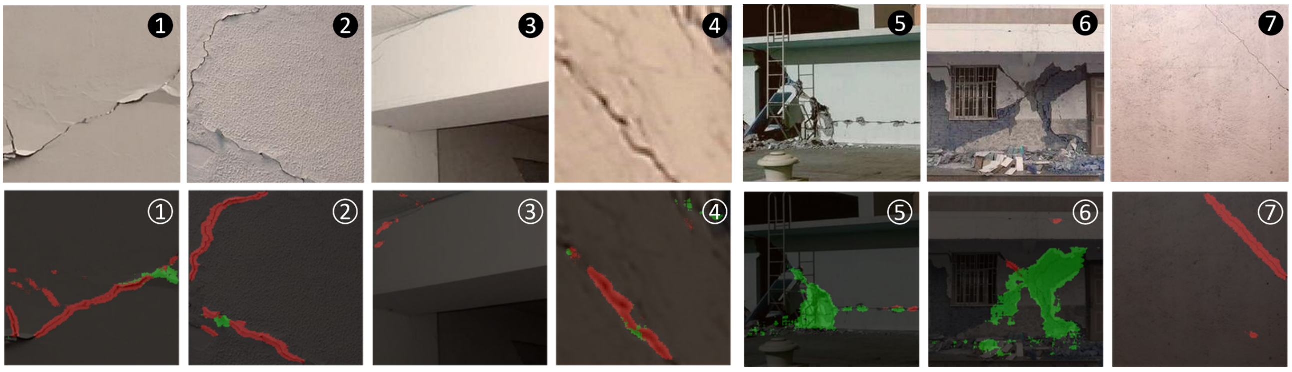

*Figure 4: Prediction results of test dataset*

For more complex object-level images, the model displays good performance in recognising and segmenting spalling and cracks at different locations of the wall in set 5. However, the model could not recognise the spalling on the left part of the wall in set 6.

The test results suggest that the model exhibits a satisfactory ability to accurately segment concrete damage in pixel-level images, while it has some limitations in object-level images. This could be attributed to the fact that object-level images are more complex and often contain irrelevant information that can potentially interfere with the identification of damage by the model. Therefore, the performance of the model in object-level images can be influenced by factors such as image complexity and variability in the location and extent of the damage.

## BIM modelling of detected damages

### Methods for geometric feature extraction of damage

To accurately represent concrete damage in the BIM model, it is necessary to extract the geometric features of the defect from the segmentation results obtained by the algorithm. The primary challenge is to find a good compromise between a necessary simplification of the outer boundary lines of the defect and preserving the details. To achieve this, different approaches have been tested and combined.

### Damage contour extraction

An option is to use the contour of the identified defect as presented in Figure 5. A contour refers to a continuous curve or boundary that outlines the shape or edges of an object. In image processing and computer vision, a contour is a sequence of connected pixels or points that form a closed curve, describing the boundaries of objects or regions within an image. Contour analysis plays a significant role in computer vision tasks, including object detection, image segmentation, and pattern recognition.

The contour was estimated using the OpenCV (Open Source Computer Vision) library, which is widely used in various applications related to image processing, computer vision, and artificial intelligence. It provides a comprehensive set of tools, functions, and algorithms for processing and analysing visual data and also provides algorithms for contour approximation.

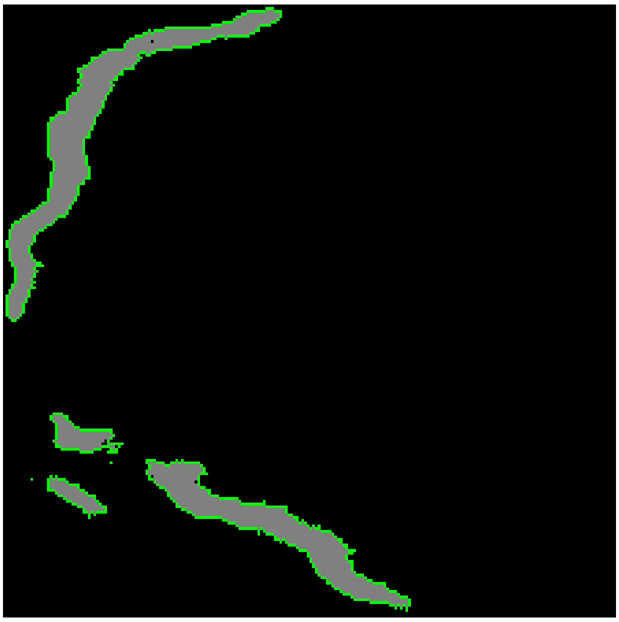
*Figure 5: Contour of the predicted result*

However, the segmented damage part may not always be continuous, resulting in numerous contours beyond the damage part and an excessive number of contour points. Attempting to extract contours directly from the segmentation results in Figure 5, six contours were retrieved instead of 4. Furthermore, even using the most simplified contour approximation algorithm, the number of contour points obtained was still close to 100, which is disadvantageous to the generation of damage in the BIM model. Hence, this method will not provide an adequate simplification way for a lightweight BIM representation.

### Convex hull of damage

To address the issue discussed, the results of the segmentation are simplified by forming a convex hull of the damage part. Calculated using the Scikit-image implementation, an open source Python library for image processing. It provides a comprehensive collection of algorithms and functions for handling and analysing digital images. Built on top of NumPy and SciPy, Scikit-image is widely used in

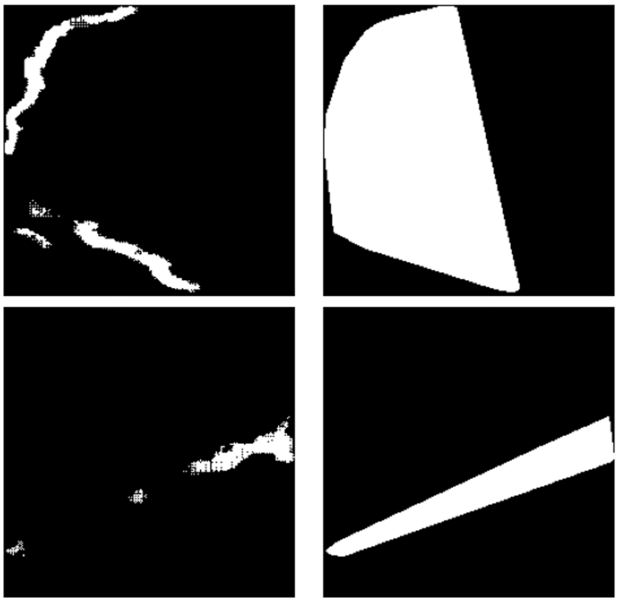

*Figure 6: Erroneous convex hull transformation. Left: original picture. Right: transformed picture.*

various domains, including computer vision and biomedical imaging.

The convex hull is a fundamental geometric concept used in various fields, including computer science and engineering. It provides a concise representation of the boundaries of a set of points. The convex hull refers to the smallest convex polygon that can cover all the given points in a two- or three-dimensional space. In 2D, the convex hull is a polygon, while in 3D, it becomes a polyhedron. A convex hull uses line segments to connect the outermost points of the shape to form a contour that encloses all points of the shape (Barber et al., 1996). However, since the segmented damage parts are distributed in various positions in the image and most of them are discontinuous, direct estimation of the convex hull in the predicted images leads to poor accuracy, as shown in Figure 6. Depending on the original image, it is likely that the convex hull will cover a larger area ignoring the details. In Figure 6, the area covered by the convex hull is much larger than the damage area of the real structure, leading to a huge error in the extraction of geometric features. Compared with direct contour extraction, the convex hull method is too rough. Therefore, the convex hull is inappropriate for direct use in images.

To address this issue, segmented damage images are evenly split into smaller images with a resolution of 28×28, resulting in a total of 64 images. For each of these smaller images, the convex hulls are generated independently. Generating convex hulls in smaller images can effectively improve the accuracy of geometric feature extraction and avoid errors caused by discontinuities in segmented parts. Subsequently, the convex hulls for all of the small images are combined to form the final convex hull result.

Figure 7 displays the process and the result. The results obtained in this way have better performance in repairing the structure than when applying convex hulls directly to

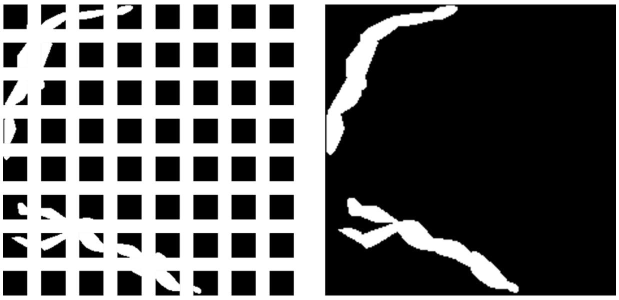

*Figure 7: Erroneous convex hull transformation. Left: original picture. Right: transformed picture.*

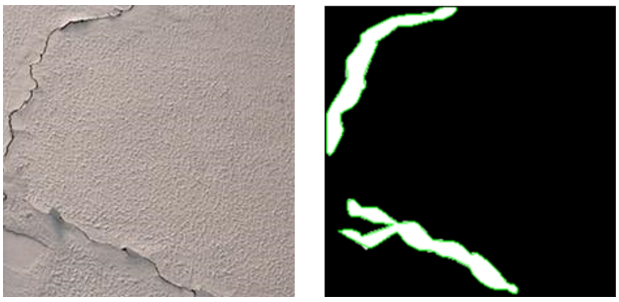

*Figure 8: Damage contour extraction.*

the images. In this study, the images are split into 64 small images. For higher accuracy, the images can be split into more and smaller images to generate convex hulls.

Although the final result may not accurately represent the exact form of the crack, it is still capable of providing an approximate representation of the position and geometric characteristics of the crack, which is beneficial for the subsequent extraction of geometric features. With this proposed method, a good degree of simplification is achieved while preserving the details necessary to analyse the defect further.

**Combination of contour and convex hull for BIM modelling**

To extract the geometric features of the defect, the methods of convex hull and contour are combined. After the images are split and the convex hull is applied, the contour of the results obtained is generated. Since the damage part becomes continuous, the number of generated contours is reduced and the geometric features of the damage are effectively presented. In addition, the number of contour points is reduced to 80, which is beneficial to reconstructing damage in the BIM model. The results of geometric feature extraction and the original image are shown in Figure 8. With the algorithm provided by Scikit-image, it is also possible to calculate the area and perimeter of the damage part.

The position of the pixels must be transferred to the local coordinates. Since no scaling factor is provided with the images in the dataset, the scale factor to translate the pixel position into coordinates is estimated at 2.7. The coordinates of each point of the contour are extracted in Python and imported into an IFC file in text form.

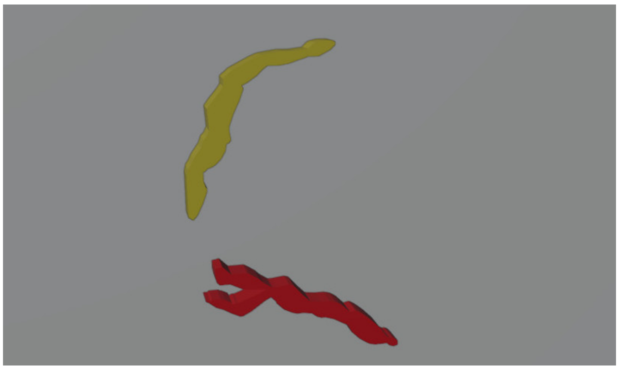

*Figure 9: BIM object of detected crack.*

The defects are transferred to a BIM using IfcOpenShell[3]. As there are no classes for damages such as crack and spalling in the IFC schema, the generic class IfcBuildingElementProxy is used. If required, a property set or quantity set containing the area or the perimeter of the defect can be added. As a geometric representation, an IfcExtrudeAreaSolid with a polyline following the outbound coordinates of the respective damage, and a thickness of 100 mm is used. The modelled damage objects are presented in Figure 9, with a wall as reference. It is obvious that the concrete cracks imported into the BIM model through the IFC schema are consistent with the segmentation prediction results.

To summarise, the automated process developed in this study for integrating the segmentation results into Building Information Modelling (BIM) involves the following steps. First, the image of the segmentation result is split, then the corresponding convex hull is generated on the small images, after that the small images are spliced to generate the contour of the convex hull, and finally the contour data is converted into an IFC file and imported into the BIM model.

## Conclusion

Existing concrete structures will deteriorate, especially when exposed to influences such as water, salt, and frost. The deterioration will cause surface defects such as cracks and spalling. Maintenance and monitoring of concrete structures are economically costly. However, with the promotion of digitalisation of construction in recent years, BIM has been playing an important role in the digitalization of construction processes and management. Hence, the focus of this study is how to integrate concrete structure damage into BIM models through the IFC schema.

In the presented study, the process of data collection and annotation is first described. In addition, a modified U-Net model is developed to implement image segmentation of concrete structure damage. Here, the transfer learning technique is applied. The model achieves an average accuracy of 94.2 % on the validation images and good results in the unseen test images. The great potential of deep learning techniques for the detection of structure damage is fully demonstrated. It also represents the possibility of automating structural health monitoring in the future.

Although the model developed in this study has shown good robustness and generalisation and performed quite well on the test dataset, there are still some shortcomings and areas for improvement in the experiments. The scarcity of the data set that contains only 2192 training images could be enhanced to improve the capabilities of the segmentation model. Although quite good performance could be achieved on pixel-level images, further improvements are necessary on object-level images, as object-level images often contain more distracting and complex information that poses challenges for accurate segmentation. Regarding data annotation, it must be noted that the images used in this study were selected and annotated by a single individual. However, in general, image data selection for training in image segmentation tasks should involve evaluation by two to three professionals with relevant expertise. Similarly, image annotation should be performed by one or two professionals, followed by a review by some experts to ensure the accuracy of the annotation. Finally, it is important to emphasise that the main objective of this study was to demonstrate the potential of deep learning techniques for the segmentation of damage to concrete structures even with a small data set and its subsequent integration into BIM modelling. Therefore, the modified U-Net implemented in this study was not exhaustively optimised and its complexity was limited.

Future research could focus on fine-tuning the network hyperparameters for improved performance or utilising more sophisticated network architectures or algorithms, such as R-CNN or Transformer networks, to achieve higher levels of accuracy. This research can be expected to contribute significantly to the field of concrete structure damage segmentation and BIM modelling. Pixel accuracy is utilised as a fundamental accuracy metric in the present study, with the option of incorporating additional accuracy metrics in subsequent studies to enhance confidence and achieve a more comprehensive assessment.

This study also proposes a method to extract geometric damage features in the prediction results, which achieves a balance between simplification and preservation of detail. The method first splits the segmented image into smaller images, then generates convex hulls of the smaller images, and finally combines the smaller images to draw the contour of the damage part. In the final part of this study, a 3D BIM representation of the geometric damage features is successfully created in the open IFC format to maximise interoperability.

AI technology has already made a lot of contributions in the field of structural health monitoring (SHM). Further integration of AI technology with Building Information Modeling (BIM) holds the promise of enhancing management efficiency and reducing time and economic costs, which play a crucial role in sustainable development and energy conservation in the future. The results of this study

[3] `https://ifcopenshell.org/`

have demonstrated the potential of deep learning in structural health monitoring and provided a feasible solution for integrating structural health monitoring and BIM into a workflow. The objective of this research was to develop a system that could automatically generate relevant IFC files and display them in BIM models using only images of concrete structures with damage. Subsequent research endeavours will focus on the refinement of the workflow to enhance accuracy and automation along with a creating larger training datasets and implementing more sophisticated data augmentation techniques.

## Acknowledgements

This work is part of the HumanTech project and has received funding from the European Union under Grant Agreement No. 101058236. HumanTech is a three-year project that started on 1 June 2022 and terminates on 31 May 2025.

## References

Artus, M., Alabassy, M. S. H., and Koch, C. (2022). A bim based framework for damage segmentation, modeling, and visualization using ifc. Applied Sciences, 12(6):2772.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. ACM Trans. Math. Softw., 22(4):469–483.

Chen, J., Lu, W., Ghansah, F. A., and Peng, Z. (2022). Defect digital twinning: A technical framework to integrate robotics, ai and bim for facility management and renovation. IOP Conference Series: Earth and Environmental Science, 1101(2):022041.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09.

Dung, C. V. and Le Anh, D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 99:52–58.

Escalona, U., Arce, F., Zamora, E., and Sossa Azuela, J. H. (2019). Fully convolutional networks for automatic pavement crack segmentation. Computación y Sistemas, 23(2).

Gao, Y. and Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. Computer-Aided Civil and Infrastructure Engineering, 33(9):748–768.

Jierong Cheng, Wei Xiong, Wenyu Chen, Ying Gu, and Yusha Li (2018). Proceedings of TENCON 2018: 2018 IEEE Region 10 Conference : Jeju, Korea, 28-31 October 2018. IEEE, Piscataway, NJ.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Li, S. and Zhao, X. (2019). Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Advances in Civil Engineering, 2019:1–12.

Li, S., Zhao, X., and Zhou, G. (2019). Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering, 34(7):616–634.

Motamedi, A. and N-Yabuki (2017). Extending bim to include defects and degradations of buildings and infrastructure facilities. The 3rd International Conference on Civil and Building Engineering Informatics in conjunction with 2017 Conference on Computer Applications in Civil and Hydraulic Engineering, Taipei, Taiwan.

Musella, C., Serra, M., Menna, C., and Asprone, D. (2021). Building information modeling and artificial intelligence : Advanced technologies for the digitalisation of seismic damage in existing buildings. Structural Concrete, 22(5):2761–2774.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1).

Sresakoolchai, J. and Kaewunruen, S. (2021). Integration of building information modeling and machine learning for railway defect localization. IEEE Access, 9:166039–166047.

Su, C. and Wang, W. (2020). Concrete cracks detection using convolutional neuralnetwork based on transfer learning. Mathematical Problems in Engineering, 2020:1–10.

Teng, S., Liu, Z., Chen, G., and Cheng, L. (2021). Concrete crack detection based on well-known feature extractor model and the yolo_v2 network. Applied Sciences, 11(2):813.

Ye, X. W., Jin, T., Li, Z. X., Ma, S. Y., Ding, Y., and Ou, Y. H. (2021). Structural crack detection from benchmark data sets using pruned fully convolutional networks. Journal of Structural Engineering, 147(11).

Yuqing Gao and Khalid M. Mosalam (2020). Peer hub imagenet: A large-scale multiattribute benchmark data set of structural images. Journal of Structural Engineering, 146(10):04020198.