Experimental Mechanics (2024) 64:1369–1380
https://doi.org/10.1007/s11340-024-01098-2

RESEARCH PAPER

Enhancing Crack Detection in Critical Structures Using Machine Learning and 3D Digital Image Correlation

O. Holzmond[1] · D.C. Roache[1] · M.C Price[1] · J. L.Walters[2] · B.R Maier[3] · X. Li[1]

Received: 18 July 2023 / Accepted: 8 July 2024 / Published online: 7 August 2024

Abstract

Background Three-dimensional digital image correlation (3D-DIC) is a non-contact monitoring technique that is able to provide accurate three-dimensional strain and displacement measurements. Previous research has shown that 3D-DIC can detect micron-scale cracks in structures as they emerge; however, because 3D-DIC is an optical sensing technique, unfavorable visual conditions due to high heat, large deformations, or a significant distance between the structure and the 3D-DIC cameras can make crack detection difficult or impossible.
Objective This research aims to develop machine learning algorithms capable of detecting characteristic crack signals in these scenarios.
Methods Localized point velocities obtained via 3D-DIC were transformed into 2D color images for machine learning segmentation. A novel dataset processing technique was utilized to produce the training dataset, which overlayed simplistic crack analogs on top of the first 50 images from the test. Different parameters from this technique were investigated to determine their effect on the model's accuracy and sensitivity.
Results The resulting model detected the onset of significant cracking with an accuracy comparable to acoustic emissions sensors. Varying the processing parameters yielded models that could detect evidence of cracking earlier, at the cost of potentially higher false positive rates. The model also performed well on structures imaged in similar testing setups that were not included in the training dataset.
Conclusion This data processing technique enables crack detection in scenarios where acoustic emissions and other sensors cannot be used. It additionally allows processes already utilizing 3D-DIC to obtain additional information about material performance during testing or operation.

Keywords Digital image correlation · Defect detection · Machine learning · Image segmentation

Introduction

Quality assurance and structure health monitoring are essential in high-performance aerospace, healthcare, and energy generation applications. Uncertainty is introduced at various times, from a structure's start of life during manufacturing through its use in potentially harsh environments. Digital image correlation (DIC) is a non-contact optical technique that can help reduce or eliminate this uncertainty. In three-dimensional DIC [1], two cameras are placed in a stereo vision setup, focused on the same object, and a calibration is performed to determine their relative position. These cameras capture a set of images at a given interval. A full-field profile of the structure is created using the stereo set of images and the calibration. Then, this 3D topological measurement can be compared across the time dimension to measure displacement [2] and strain. Importantly, this method can measure out-of-plane displacement [1] and can achieve micron-level accuracy [1, 3, 4] in well-calibrated cases.

Comprehensive studies such as Roux et al. [5] have shown that 3D-DIC can accurately detect cracks on the micron scale in various materials. Unfortunately, the viability of

✉ X. Li
xl3p@virginia.edu

[1] Department of Mechanical and Aerospace Engineering, University of Virginia, 122 Engineer's Way, Charlottesville, VA 22903, USA

[2] Westinghouse Electric Company, Hopkins, SC 29061, USA

[3] Westinghouse Electric Company, Churchill, PA 15235, USA

3D-DIC can be limited by several factors commonly found in manufacturing, energy-intensive applications, and other applications that could greatly benefit from this crack detection. Visual noise and optical distortions can be introduced to the raw images by flashing lights [6] (such as sparks) and moving gases (such as hot air). 3D-DIC resolution can further be impacted by poor camera mounting or large structure deformations relative to the crack size.

This study aims to further enable this method by utilizing machine learning to facilitate the detection of cracks present in the 3D-DIC data, allowing for the detection of cracks not able to be seen via traditional digital image processing techniques or even by the human eye. This study developed a method utilizing data gathered during an expanding plug test [7]. Previously, structure failure and cracking during an expanding plug test have been detected via 3D-DIC by computing the hoop strain and detecting when the strain exceeds a threshold value [8–10]. This study goes a step further and identifies localized cracking throughout the test, allowing for a more precise indication at the exact moment a structure begins to fail and where in the structure this failure originates from.

Materials and Methods

Digital Image Correlation and Expanding Plug Test Setup

Two Grasshopper3 digital cameras (FLIR) were set up in a stereoscopic array approximately 200 mm from the target structure. The structure requires a high-contrast surface optical texture [11] to allow the DIC system to achieve a good correlation. Thus, the face of the structure facing the camera array was spray painted with black and white matte finish spray paint (Rust-Oleum) to achieve a non-uniform speckle pattern. Testing setup specifics can be seen below in Table 1.

The expanding plug test was performed using an Admet universal testing machine with a 10kN load cell. It mirrored the expanding plug test found in Roache et al. [9], except a displacement-regulated loading rate of 0.5 mm/min was used. An acoustic emissions (AE) sensor (Mistras 1283 AE Node) was clamped onto the upper arm of the testing machine to provide a secondary validation of crack formation times, as demonstrated in [9]. An illustration of the test setup can be seen in Fig. 1.

Table 1 DIC test setup properties

	DIC Test Setup
Technique Used	3D Digital Image Correlation
Camera	2 × FLIR Grasshopper3
Sensor	14 bit, 2048 pixels × 2448 pixels
Lens	2 × Sigma 105 mm F2.8
Imaging Distance	~ 200 mm
Field of View	32 mm × 27 mm
Pixel resolution	~ 1.3 µm/pixel

Calibration targets, along with the software package Vic3D-8 (Correlated Solutions), were used to determine the exact position of the cameras relative to each other and the structure. Each pair of stereoscopic images were divided into many overlapping regions. Due to the non-uniform speckle pattern, each region is distinct, and thus regions can be matched between the image pair even in the presence of structure deformation. By matching up these regions, a transform is computed between the two cameras, which, combined with the calibration information, can determine the exact X, Y, and Z coordinate positions of each of the regions, resulting in a 3D point cloud topography of the structure. By selecting the same regions at each time step, this topography can be compared across time to determine strain and velocity at each point.

Data Preprocessing and Model Setup

As the anticipated crack width of ~ 1 µm is less than the size of a pixel, direct observation of cracks forming in the digital images is unlikely. Instead, point velocity was monitored for a large increase in a localized area. This has the advantage of potentially having a spatially larger signal, as the crack opening also causes neighboring points to experience higher-than-normal acceleration. In turn, this can also make it more difficult to pinpoint the location and size of every crack – especially if the DIC acquisition time is too long. The DIC correlation data was used to compute the X displacement and change in X velocity of each point. The X, Y, and change in X velocity for each point were then interpolated into a regular 224 pixel by 224 pixel grid, with the value at each pixel representing the change in X velocity. These 2D images were then able to be used in traditional image-based machine learning processes.

To allow for easier adoption, self-supervised learning [12, 13] was used for model training. This allows for the use of datasets with little to no human labels. This is especially important as identifying cracks quickly in the field can be non-trivial and require previously failed structures to create labeled data from. Instead, the training dataset for the model is generated through the systematic creation of simplified simulated cracks on a subset of initial images.

To create images with simulated cracks, a set of seed images consisting of the first 50 images captured by the DIC system was used as a background to create the cracks on. These were chosen as we were confident that these images would be at a load too small to exhibit any cracking but would have a good sample of the noise encountered

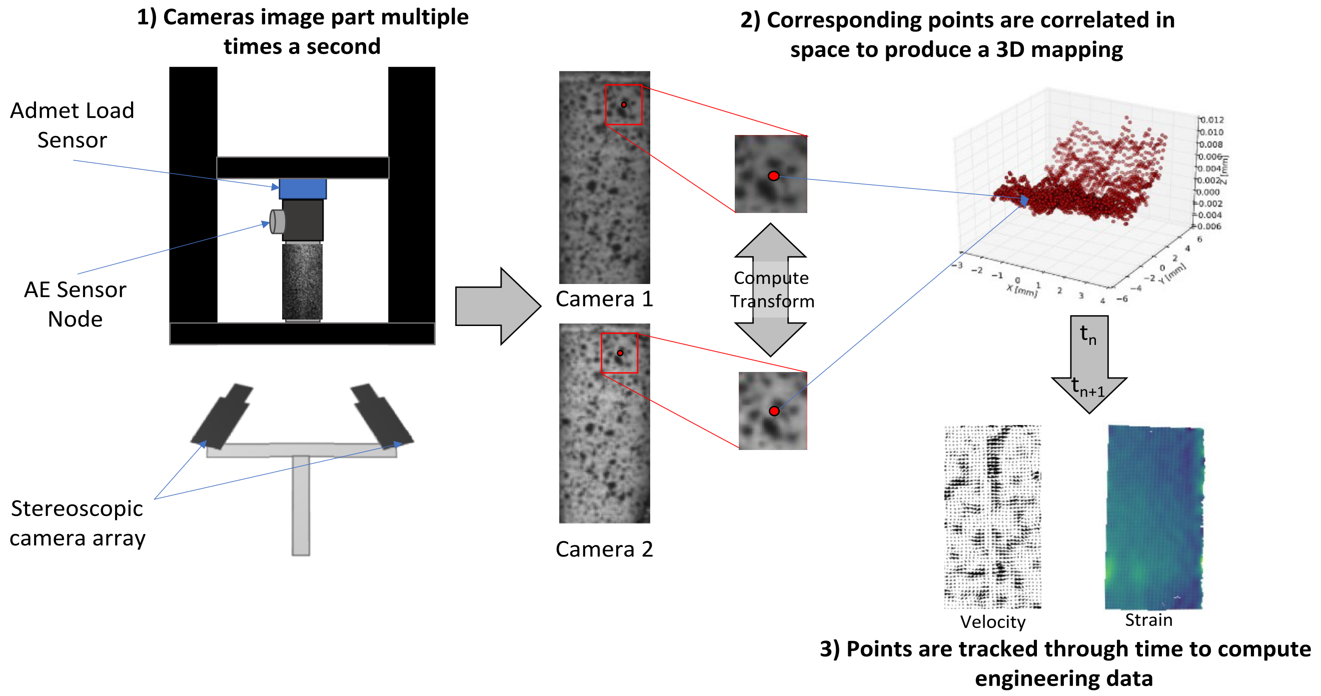
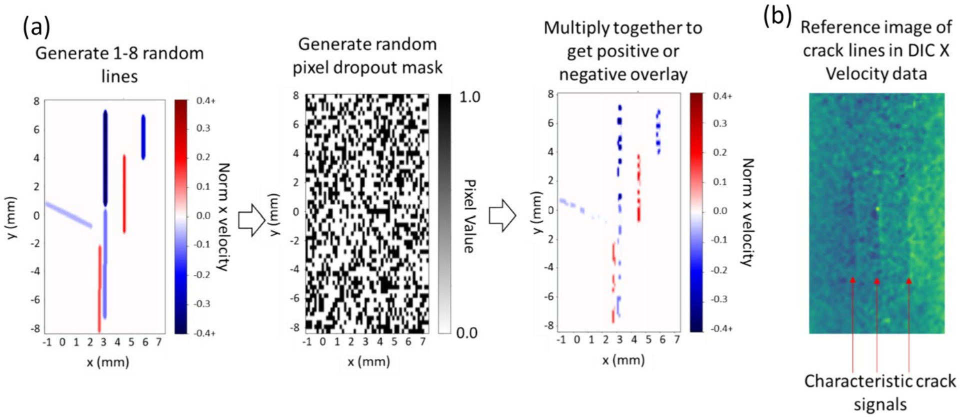

1) Cameras image part multiple times a second
Admet Load Sensor
AE Sensor Node
Stereoscopic camera array
Camera 1
Camera 2
Compute Transform
2) Corresponding points are correlated in space to produce a 3D mapping
X [mm]
Y [mm]
Z [mm]
t_n
t_n+1
Velocity
Strain
3) Points are tracked through time to compute engineering data

Fig. 1 Expanding plug test and instrumentation setup

during the test. This is important because any cracks initially present in the seed images will not be flagged as valid cracks for the model to be trained on, and thus the model could inadvertently learn to ignore some types of cracks. To create the simulated cracks, 1 to 8 random lines were created using random width, length, and magnitude values within predefined ranges. The crack analogs to be used in the creation of the training set were designed to be as simple as possible to allow for the model to be as generic as possible, and to minimize the amount of technical information needed to be able to implement the model in a production environment. The choice of straight lines was informed by the specific experimental setup and anticipated failure mode (namely, hoop stress resulting in long vertical cracking). In other scenarios where more complex curving cracks are anticipated, modifications to analog primitives may be required to enable effective crack detection.

However, simply using the lines as drawn was not effective. This is due to the fact that basic lines were not representative of the characteristic crack signals (see Fig. 2(b)) that the model is being designed to eventually detect. To compensate for this, a pixel dropout mask generated from random-sized noise was created to apply to the drawn lines, causing part of the lines to disappear. Then they were multiplied by a mask generated from random-sized noise to provide pixel dropout (see Fig. 3). This process is illustrated in Fig. 2.

Fig. 2 (**a**) Positive example generation with random-sized pixel dropout (γ = 0.6) (**b**) Example of characteristic crack signals



This random sized noise is generated by creating an image with smaller dimensions than the target dataset, filling the image with values sampled from a uniformly random distribution (referred to as noise), and then upscaling the image to the final target dataset dimension. This results in an image of noise where the size of each sample is potentially greater than a single pixel. To determine the dimensions of the smaller image used to initially generate the noise, a target sample pixel size was selected according to an equation reproduced below in (1).

$$\begin{aligned} x &= 1 + \|norm(\mu, \sigma)\| \\ Noise\ Size &= clip(x, a, b) \end{aligned} \tag{1}$$

Here, $clip(x, a, b)$ represents the clip function that returns a if $a > x$, b if $b < x$, and x otherwise. $norm(\mu, \sigma)$ is a function that takes a random sample from a normal distribution with mean μ and standard deviation σ. The values used in our experiments were $\mu = 2, \sigma = 1.75, a = 1, b = 18$.

The dataset image dimensions were then divided by this size value to obtain the dimensions of the smaller image. Noise was generated by sampling a uniform distribution between zero and a noise coefficient hyperparameter C_n, chosen depending on the amount of noise present in the seed images. This noise image was then scaled via nearest neighbor interpolation to the dataset image dimensions. Finally, a Gaussian blur was applied to the noise image, with a kernel size dictated by the amount of noise present in the seed dataset. In the pixel dropout mask generation, this noise was then passed through a threshold of $\frac{\gamma}{C_n}$ where all pixels less than this value were set to zero, and all others were set to one.

After the first set of lines was generated and applied, a second set of lines was generated in a similar fashion. However, these lines possessed characteristics distinct from the initial set, with the range of possible values for length and width different from the first set. This second set of lines served as a set of counter-examples for the model.

After the two sets of lines were added to the initial seed image, random-sized noise was applied to the entire image. This noise was generated in the same manner as for the pixel mask dropout, except no thresholding was performed. This was done to help prevent overfitting due to the small number of original input images, as well as to provide more obfuscation of the lines we are training the model to detect. Finally, augmentation techniques commonly employed in image processing, including rotation, translation, horizontal flipping, and scaling, were applied. This overall process is demonstrated in Fig. 4.

The Pytorch segmentation library [14] was used as the underlying framework to construct the prediction model architecture. Multiple models including MANet [15], UNet [17] and UNet++ [18] were canvassed along with multiple pre-trained encoders (Inceptionv4 [16], Resnet [19], VGG [20], DPN [21], ResNeXt [22], and ResNeSt [23]) to determine the best model-encoder combination.

In the initial testing, all of the models and encoders used were able to yield acceptable results given proper hyperparameter tuning, and none stood out as exceptionally good or bad compared to the others. The combination of a MANet segmentation model with an Inceptionv4 encoder yielded the best qualitative results in detecting known cracks and was subsequently used for all tests. Loss for each prediction was computed by taking the pixel-wise mean squared error between the predicted mask ($\hat{Y}$) and the positive example mask (Y). An illustration of the training loop is shown in Fig. 5.

In evaluating the performance of different model configurations, the following criteria were investigated:

1. The ability of the model to accurately detect the artificial line analogs present in the augmented training set, without false positives,
2. The ability of the model to not falsely detect cracks in the initial 50 seed images known not to have any cracking,
3. The ability of the model to not falsely detect cracks in other images that were known not to have cracking, but

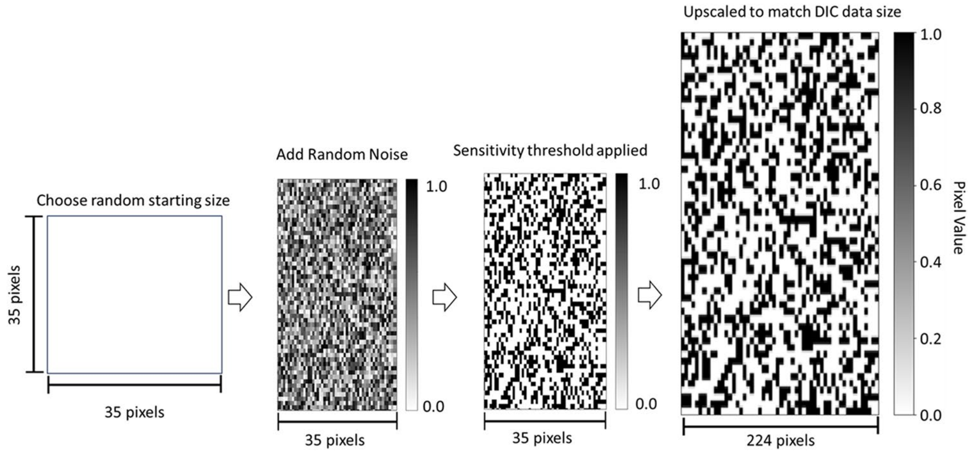


Fig. 3 Random sized noise generation process

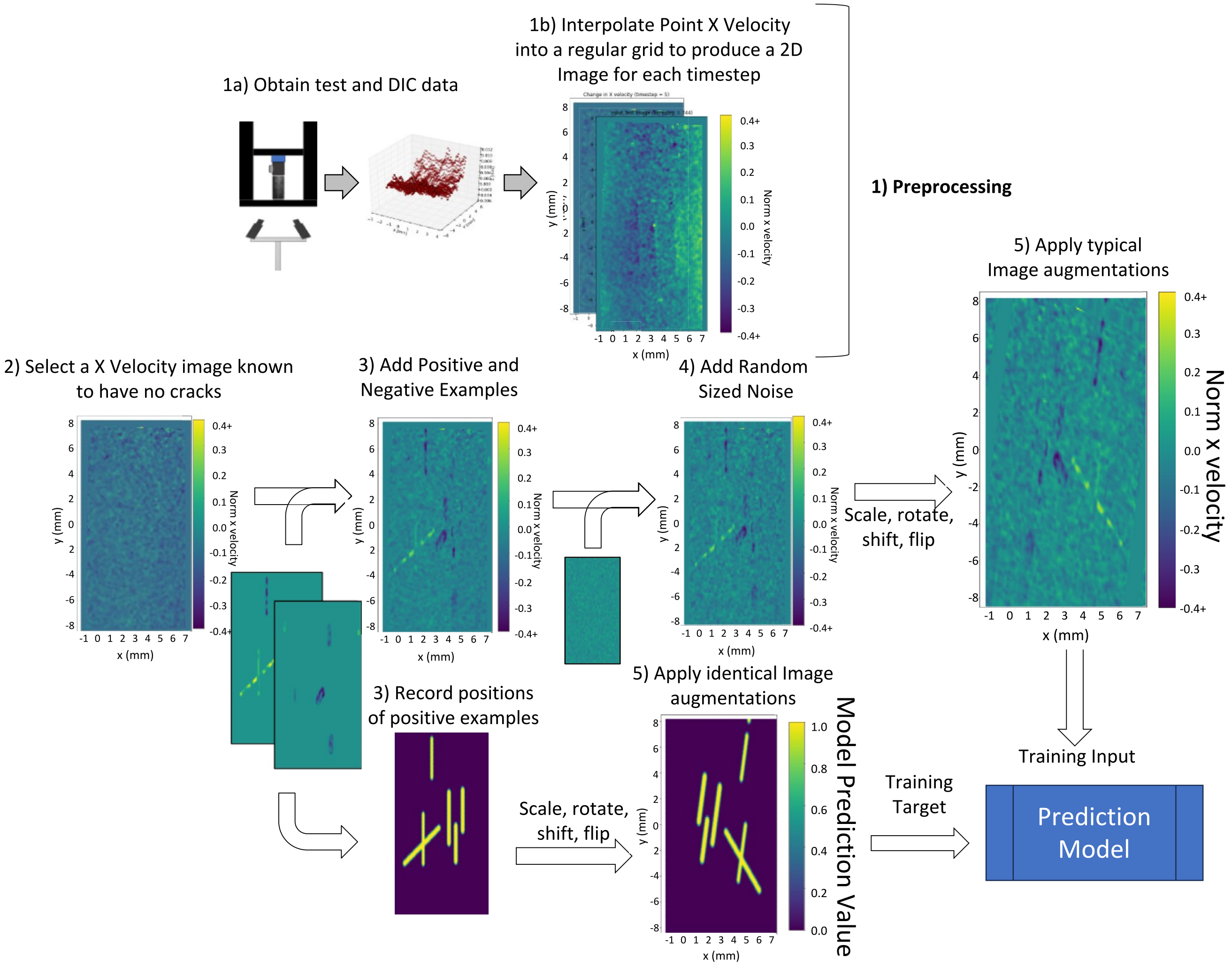

1a) Obtain test and DIC data
1b) Interpolate Point X Velocity into a regular grid to produce a 2D Image for each timestep
1) Preprocessing
2) Select a X Velocity image known to have no cracks
3) Add Positive and Negative Examples
4) Add Random Sized Noise
5) Apply typical Image augmentations
Scale, rotate, shift, flip
3) Record positions of positive examples
Scale, rotate, shift, flip
5) Apply identical Image augmentations
Training Target
Training Input
Prediction Model
Norm x velocity
Model Prediction Value
y (mm)
x (mm)
0.4+
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4+
1.0
0.8
0.6
0.4
0.2
0.0

Fig. 4 Input augmentation process

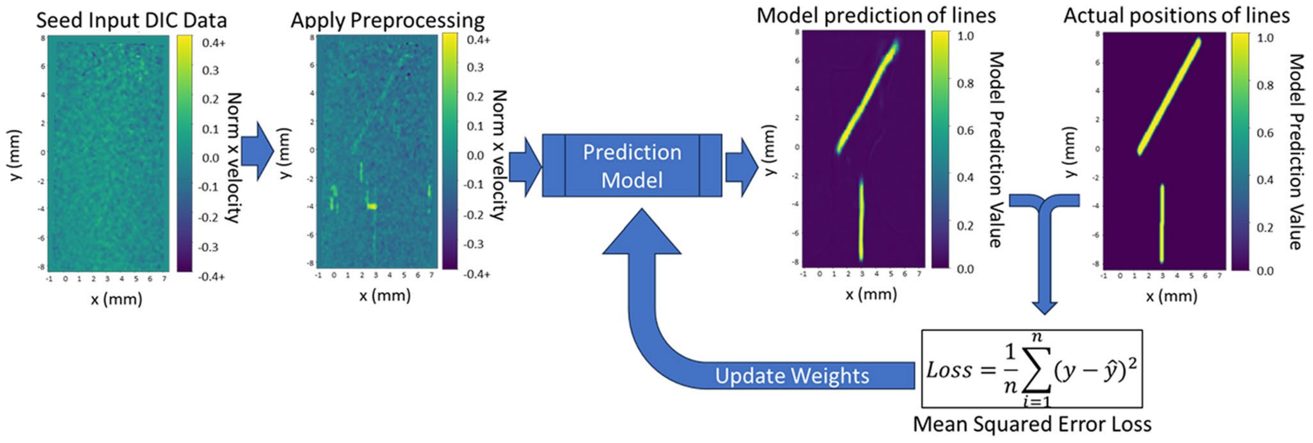


Fig. 5 Prediction model training loop

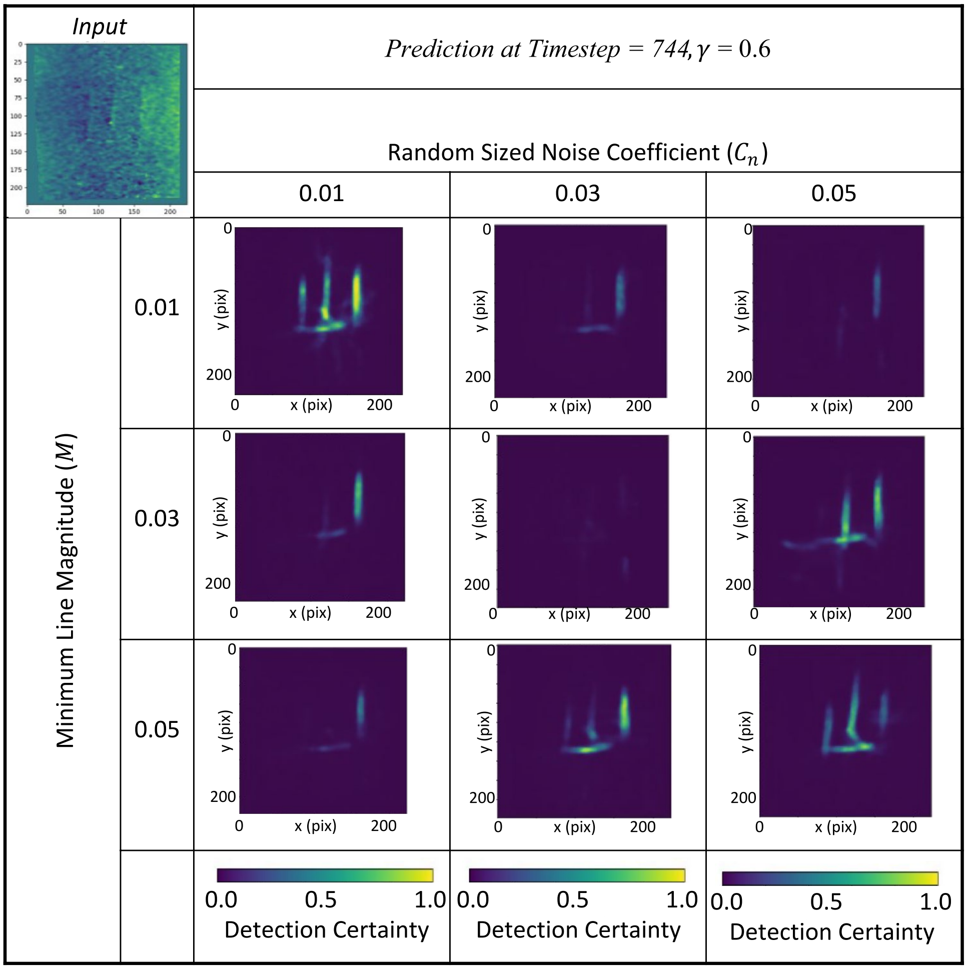


Fig. 6 Array of predictions with various random-sized noise coefficients and minimum line magnitude values (x and y axis are shown in pixels for easier viewing)

not part of the seed image set (in our experiment, this comprised of the images taken at DIC timesteps 50-99),

4. The ability of the model to detect cracks in test images that are known to have cracks (such as those immediately leading up to the failure of the structure),
5. The model's agreement with data obtained from the acoustic emissions and load sensors.

The three primary hyperparameters that were varied were: The threshold amount used while generating the random-sized noise pixel dropout of generated lines (γ), the noise coefficient used while overlaying noise to the input image (C_n), and the minimum magnitude used while drawing the example lines (M). Other hyperparameters investigated include the initial learning rate for the Adam [24] optimizer and the total number of epochs the model was trained for. Learning rate values of 10^{-3} to 10^{-4} tended to yield good results, and total epochs varied between 50 and 500 epochs depending on the other hyperparameters.

Results and Discussion

An array of tests were conducted to discover the potential impact each hyperparameter would have on detection rates. Figure 6 shows results from predictions for DIC timestep 744 with $\gamma = 0.6$ and C_n and M equaling one of 0.01, 0.03, or 0.05. The spatial axes have been changed from previous illustrations to display pixels instead of mm for easier viewing, but they represent the same data.

The C_n and M hyperparameters have a significant relationship, making the effect of each dependent on the other. As shown in Fig. 6, holding $C_n = 0.01$ and increasing M decreases the quality of the crack detection. The opposite is true when holding $C_n = 0.05$ and increasing M. This could suggest that there is an optimal clarity of lines, where the model must learn to detect line type objects, without learning to only detect perfect line segments.

Figure 7 shows the effect of increasing the minimum line magnitude while holding all other hyperparameters constant on the augmented training dataset. While the trend in Fig. 6

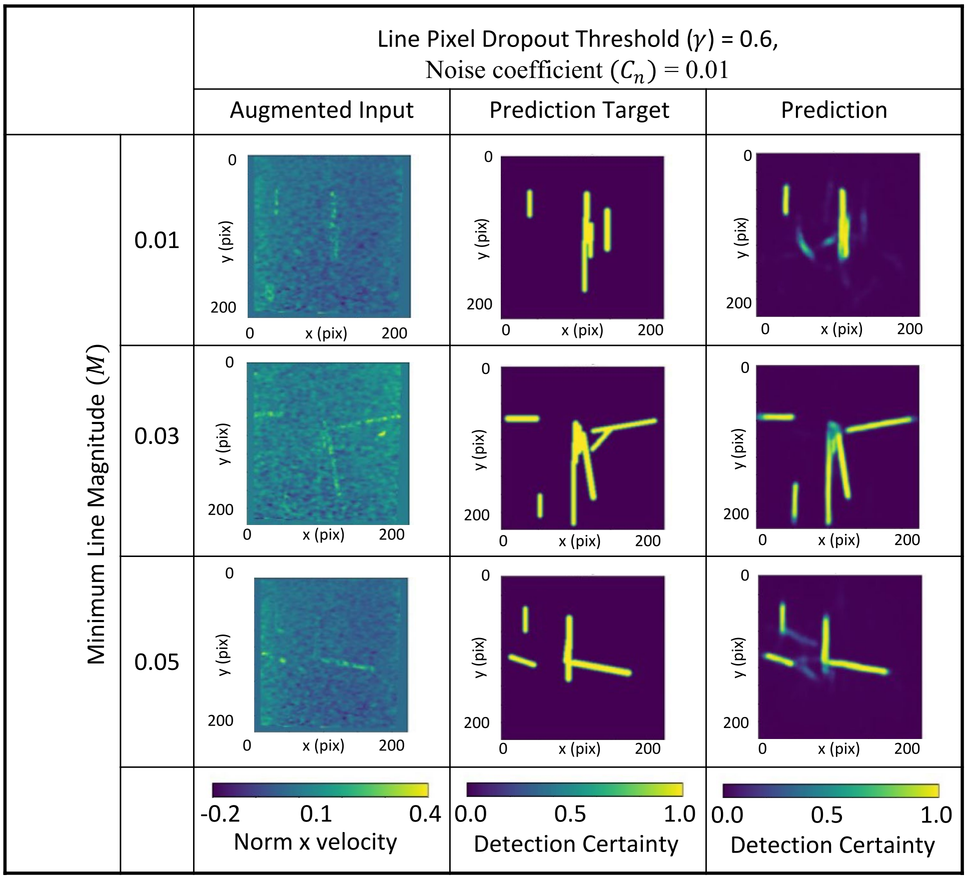


Fig. 7 Training input, prediction target, and prediction for line pixel dropout threshold = 0.6, noise coefficient = 0.01, and minimum line magnitude = 0.01, 0.03, and 0.05

for $C_n = 0.01$ shows that increasing minimum line magnitude decreases detection certainty, the trend in Fig. 7 shows significantly less change. In isolation this could be evidence that the crack signals correspond to line magnitudes on average that are closer to 0.01 than 0.05, however, the previously mentioned trend for Fig. 6 in the case of $C_n = 0.05$ does not support this conclusion. Alternatively, this could be evidence that with low noise coefficients and high minimum line magnitude, the model may start to align too closely towards the simplistic line analogues (and for example, start looking for the perfectly rounded ends of the line segments), and thus become less reliable for detecting more abstract line type shapes (such as those that we anticipate for cracks).

In the $C_n = 0.05$ case for Fig. 6, the lowest line magnitude scenario ($M = 0.01$) contains noise that is five times the minimum line magnitude. This could result in training input – target pairs where the target lines are mostly or completely obscured by noise, resulting in impossible to predict scenarios. This could have the effect of causing the model to have many false positives as it tries to match non-existent signals, or for the model to effectively be trained on less data. As the minimum line magnitude increases, the signal to noise ratio would increase, reducing these occurrences, and providing better detection certainty, which is consistent with the seen results.

The predictor had the poorest performance in Fig. 6 at $C_n = M = 0.03$, which seems not to follow the previously mentioned trends. This could suggest another factor at play, potentially concerning the size or amount of noise inherent in the data.

The configuration that was the most consistent with the five evaluation criteria was $\gamma = 0.5, C_n = 0.03, M = 0.05$. Figure 8 demonstrates the model's ability to accurately detect cracking in non-seed, non-augmented images (criteria 4) while not having large amounts of false positives in the preceding and subsequent images.

Multiplying the prediction mask with the input image creates a detection image for each timestep. This represents the detected cracks, as well as their magnitude. By summing each of these detection images, a plot is created which gives an overview of when in the process significant cracking occurs. A plot for the detection sums for the balanced model configuration can be seen in Fig. 9.

Referring back to the previously stated performance criteria, we see that the model displays little to no false positives on images from the seed dataset (DIC timesteps 0 – 49) and images that are not expected to have cracking, and also not part of the seed dataset (DIC timesteps 50–99), satisfying criteria 2 and 3.

Fig. 8 Results from balanced predictor configuration, predicting crack locations at the timestep before, during, and after cracks

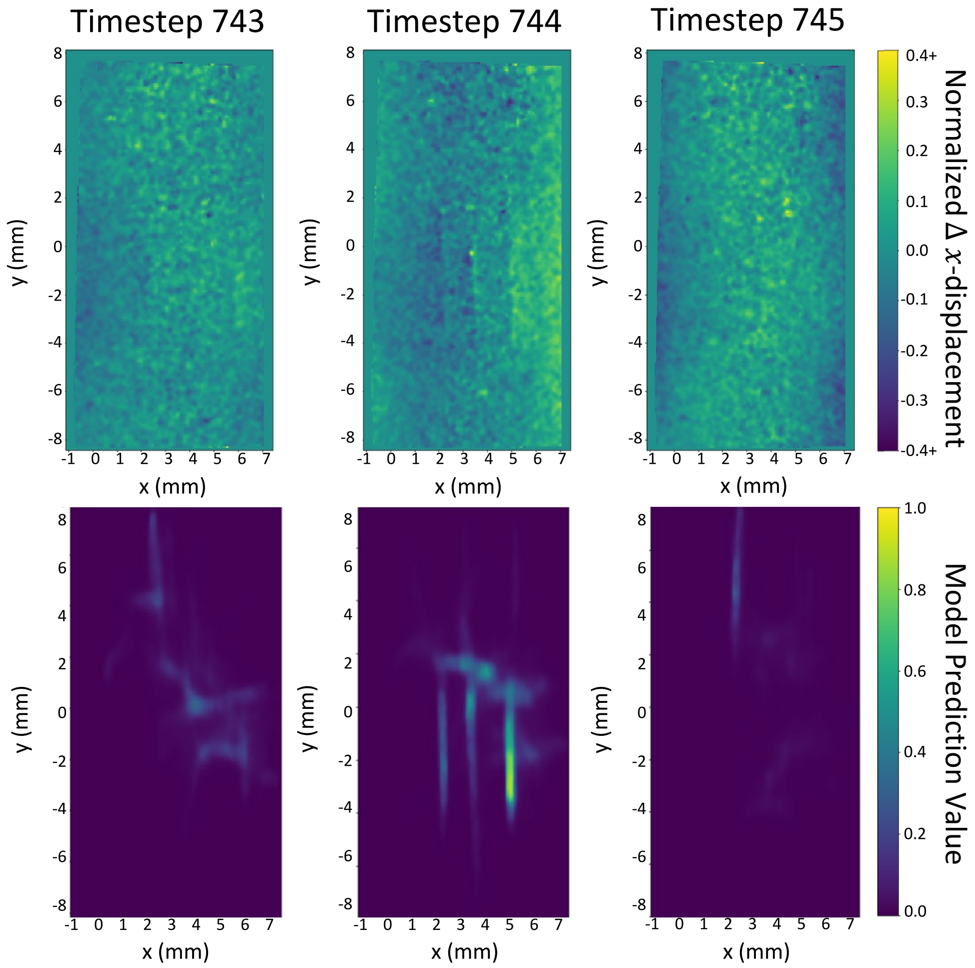
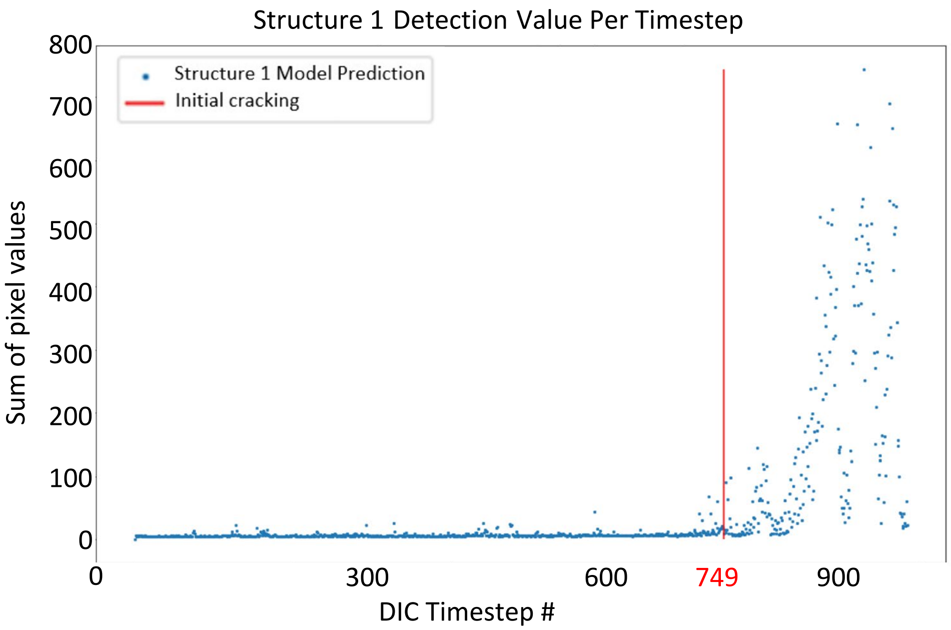


Fig. 9 Detection image sum for each timestep for the balanced model configuration



The acoustic emissions data obtained during the test (Fig. 10) is evaluated and compared against this detection value plot to determine the prediction model's ability to detect cracks accurately (criteria 5). Both plots indicate that significant cracking starts at approximately timestep 749 (shown as a vertical red line). Additionally, both plots show small detections around timestep 450. This increases confidence that the model could serve as a primary detection or complement other sensors as a detection validation mechanism. Additionally, it is apparent from Fig. 10 that the AE setup displays a larger amount of random noise compared to that of proposed model as shown in Fig. 9. Depending on the specific environment, this could cause either the AE sensor setup, or the proposed model to detect cracking earlier. A lower threshold applied to the structure 1 AE signal could've detected unwanted part loading as shown as a slight increase in AE signal slightly after the 600 DIC timestep, allowing earlier warning of a potential fault state. However, this signal is not always present (as is the case in Fig. 13), and can be signs of non-critical loading, such as those around the 300 DIC timestep. Likewise, the deformation of the speckle coating may appear as characteristic signals that the proposed model is able to detect but not the AE sensor, allowing for the proposed model to potentially achieve earlier detection in certain specific applications.

Changing the model's parameters to $\gamma = 0.7$, $C_n = 0.03, M = 0.01$ increased its sensitivity and allowed the model to detect some of the initial formations of the detected cracks. As seen in Fig. 11, the eventual cracking observed in DIC timestep 744 is seen in timestep 743 (taken 0.25 s before). These cracks were previously not able to be detected. These detections could become more reliable by including more data in the model's analysis for each timestep (such as giving it information about the previous timestep or the change in point Y velocity).

In Fig. 9, the original model's detection value remains near zero for most of the test. The start of significant cracking takes the form of a slight baseline increase (around timestep #700) and higher values for a few points, eventually leading to high but varied values across all points. This is contrasted with the detection value plot for the more sensitive model in Fig. 12. In the more sensitive model, the model's detection value has a higher mean and standard deviation in the pre-cracking phase. Additionally, instead of the sharp value increase exhibited in Fig. 9 at the onset of cracking, the sensitive model's detection values seem to increase smoothly and gradually (although once cracking has started, we see a similar jump in detection value). This gradual increase could be a negative side effect of the increased sensitivity in the form of additional false positives, or it could indicate small, less significant cracks forming on the structure's surface or in the paint. However, no matter the cause, the smoother transition of values around the onset of significant cracking could increase the difficulty of detecting the onset of cracking with the sensitive model.

One major advantage of the self-supervised nature of the model is that, because a majority of the input images are synthetic noise or positive/negative examples, a model can be trained on a given structure and then later used on another structure in a similar (but not necessarily identical)

Fig. 10 AE normalized energy versus DIC timestep

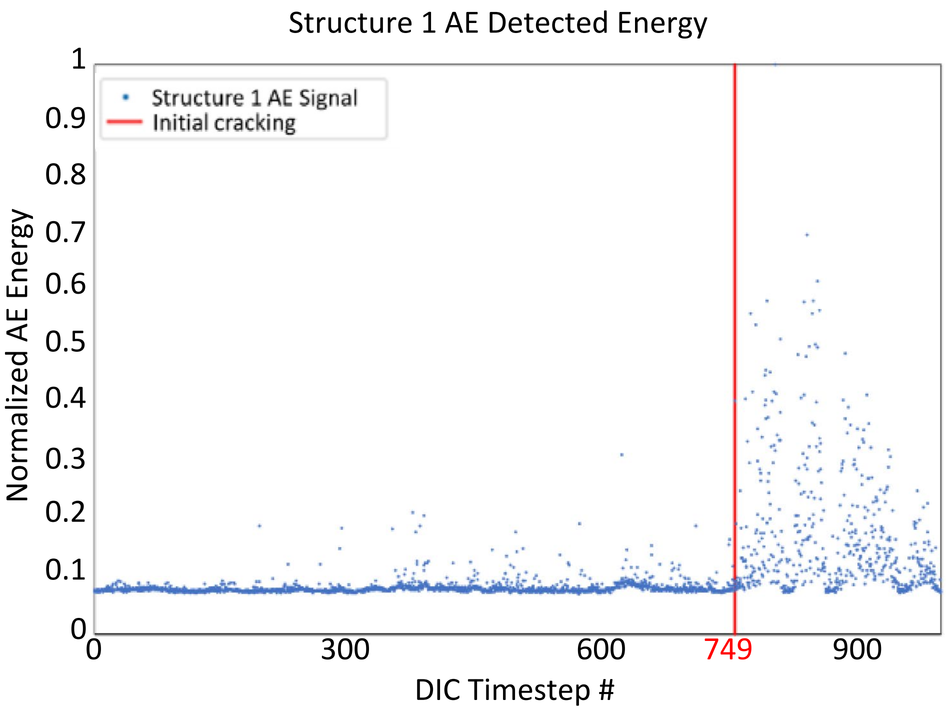


Fig. 11 More sensitive configuration with early pre-crack detection

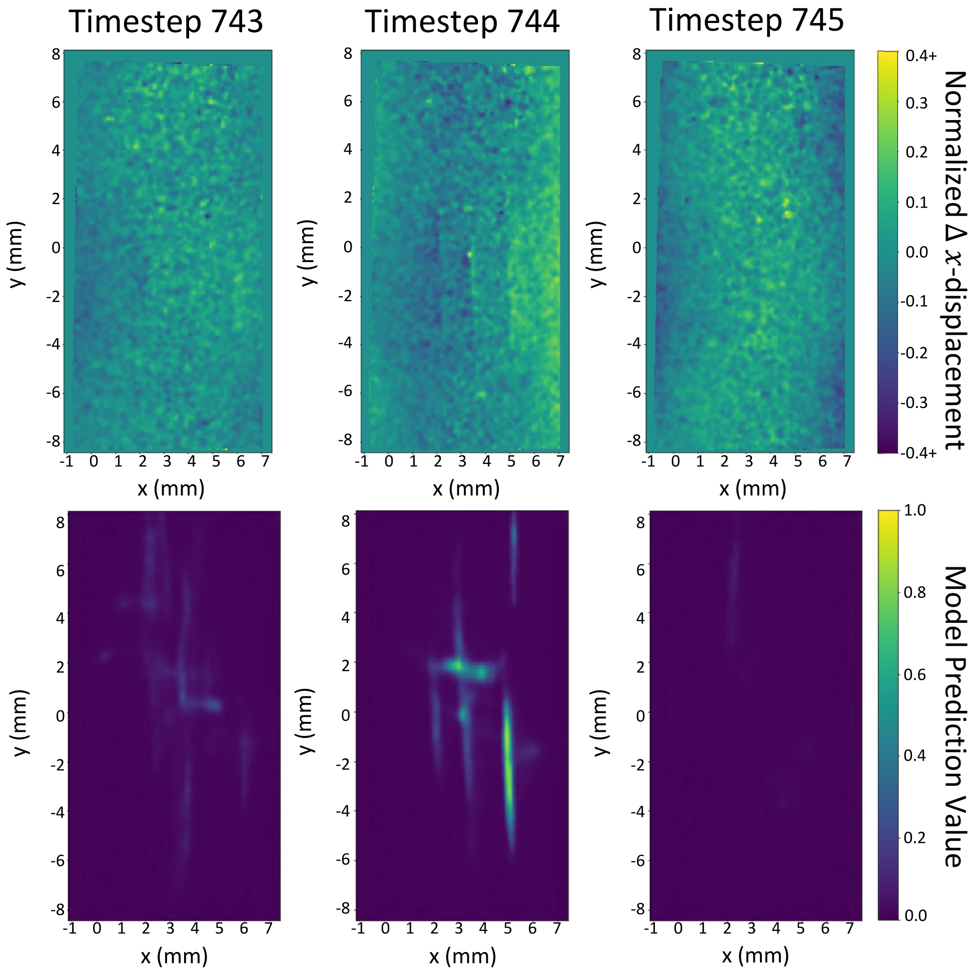
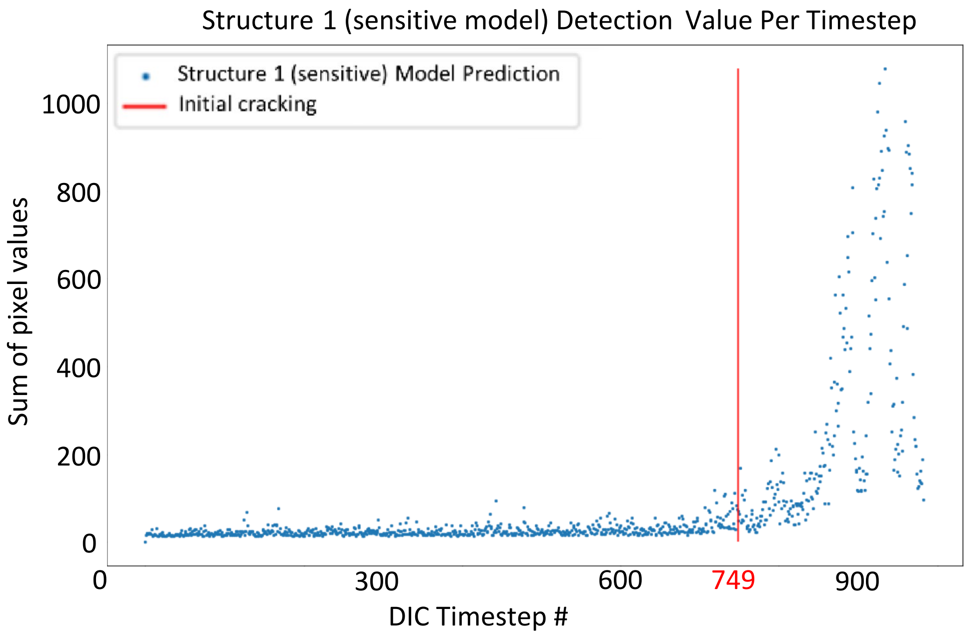


Fig. 12 Detection image sum for higher sensitivity model



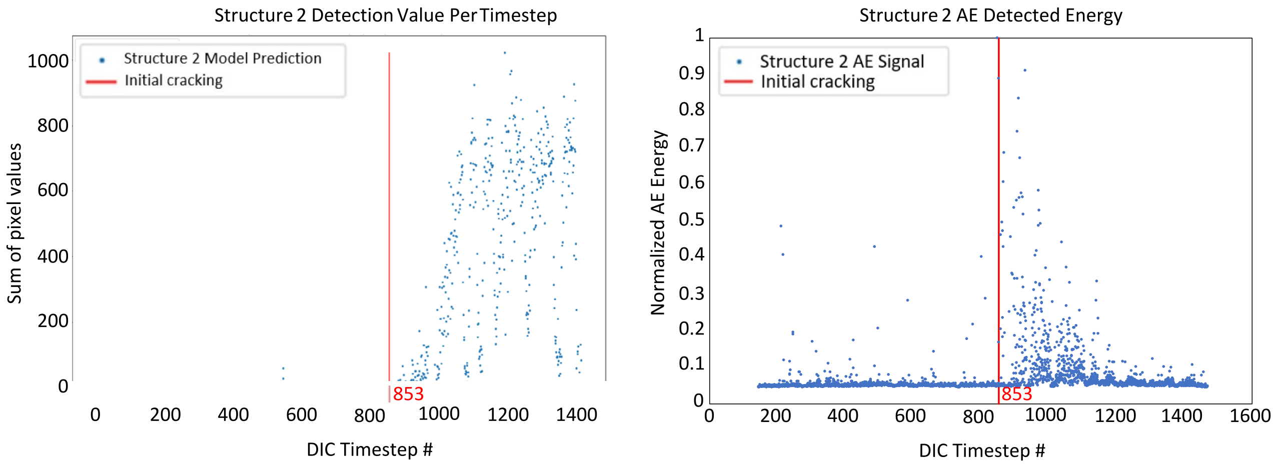


Fig. 13 AE and model prediction plots for Structure #2

environment. To test this, the model previously trained with $\gamma = 0.7, C_n = 0.03, M = 0.01$ was used without additional training to detect cracking in a different structure (referred to as structure 2). Figure 13 contains the detection value plot and AE energy plot for structure 2. Despite not being trained on any data from structure 2, the model's detection value plot and AE plot both experienced a jump at around timestep 853 (shown as a red vertical line). In contrast to the results obtained from testing on the original structure (referred to as structure 1), the detection plot seems to only increase at this value instead of slightly before it. This difference could be due to modifications in the testing environments, or the original model might not have been sensitive enough for structure 2 in general. The inaccuracies encountered when transferring the model to new structures could potentially be remedied by fine-tuning the model with some of structure 2's initial testing data or using different hyperparameters.

Conclusion

A self-supervised model was created using existing machine learning segmentation methods and novel data preprocessing to detect cracking in structures detected via digital image correlation under non-ideal conditions. This method does not need human-labeled data other than a small dataset known not to have cracks, enhancing its adaptability and making it more viable as a field technique that can be set up without expensive equipment or significant labor. Additionally, the model's sensitivity can be adjusted based on some preprocessing parameters, allowing users to tailor the model for their specific application. The model also demonstrated acceptable performance on data from structures other than the one initially trained on, decreasing the time and effort required for implementation.

Research is still needed to better understand the role of different hyperparameters, such as minimum line magnitude, pixel dropout percent, and random-sized noise coefficient, on the model's sensitivity. Future experiments involving fine-tuning the model for new structures could also show increased performance. Additionally, testing done in a manner that would allow the DIC data to be directly compared to data from an SEM or optical microscope would shine significant light on this method's exact accuracy and capabilities.

Funding The material is based upon work supported by the Department of Energy under Award Number DE-NE0009033. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Declarations

Conflict of Interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. McNiel SR, Helm JD, Sutton MA et al (1996) Improved three-dimensional image correlation for surface displacement measurement. Soc Photo-Optical Instrum Eng 35(7):1911–1920. https://doi.org/10.1117/1.600624

2. Sutton M, Wolters W, Peters W, Ranson W, McNeill S (1983) Determination of displacements using an improved digital correlation method. Image Vis Comput 1(3):133–139. https://doi.org/10.1016/0262-8856(83)90064-1
3. Krehbiel JD, Lambros J, Viator JA, Sottos NR (2010) Digital image correlation for improved detection of basal cell carcinoma. Exp Mech 50(6):813–824. https://doi.org/10.1007/s11340-009-9324-8
4. Vend Roux G, Knauss WG (1998) Submicron deformation field measurements: Part 2. Improved digital image correlation. Exp Mech 38(2):86–92. https://doi.org/10.1007/BF02321649
5. Roux S, Réthoré J, Hild F (2009) Digital image correlation and fracture: an advanced technique for estimating stress intensity factors of 2D and 3D cracks. J Phys D Appl Phys 42(21):214004. https://doi.org/10.1088/0022-3727/42/21/214004
6. Lecompte D et al (2006) Quality assessment of speckle patterns for digital image correlation. Opt Lasers Eng 44(11):1132–1145. https://doi.org/10.1016/j.optlaseng.2005.10.004
7. Jiang H, Wang JAJ (2018) Development of cone-wedge-ring-expansion test to evaluate the tensile HOOP properties of nuclear fuel cladding. Prog Nucl Energy 108:372–380. https://doi.org/10.1016/j.pnucene.2018.06.015
8. Roache DC et al (2022) Unveiling damage mechanisms of chromium-coated zirconium-based fuel claddings at LWR operating temperature by in-situ digital image correlation. Surf Coatings Technol 429:127909. https://doi.org/10.1016/j.surfcoat.2021.127909
9. Roache DC et al (2020) Unveiling damage mechanisms of chromium-coated zirconium-based fuel claddings by coupling digital image correlation and acoustic emission. Mater Sci Eng A. https://doi.org/10.1016/j.msea.2019.138850
10. Heim FM, Daspit JT, Holzmond OB, Croom BP, Li X (2020) Analysis of tow architecture variability in biaxially braided composite tubes. Compos Part B Eng 190:107938. https://doi.org/10.1016/j.compositesb.2020.107938
11. McNeill SR, Sutton MA, Miao Z, Ma J (1997) Measurement of surface profile using digital image correlation. Exp Mech 37(1):13–20. https://doi.org/10.1007/BF02328744
12. Doersch C, Zisserman A (2017) Multi-task self-supervised visual learning. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, pp 2070–2079. https://doi.org/10.1109/ICCV.2017.226
13. Doersch C, Gupta A, Efros AA (2015) Unsupervised visual representation learning by context prediction. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, pp 1422–1430. https://doi.org/10.1109/ICCV.2015.167
14. Iakubovskii P (2019) Segmentation models pytorch. GitHub. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch
15. Xu Y, Lam H-K, Jia G (2021) MANet: A two-stage deep learning method for classification of COVID-19 from Chest X-ray images. Neurocomputing 443:96–105. https://doi.org/10.1016/j.neucom.2021.03.034
16. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-ResNet and the impact of residual connections on learning. 31st AAAI Conf Artif Intell AAAI 31(1):4278–4284. https://doi.org/10.1609/aaai.v31i1.11231
17. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. [Online]. Available: http://lmb.informatik.uni-freiburg.de/. Accessed 20 May, 2023
18. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J (2018) Unet++: A nested u-net architecture for medical image segmentation. In: Lecture Notes in Computer Science. pp 3–11. https://doi.org/10.1007/978-3-030-00889-5_1
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2016:770–778. https://doi.org/10.1109/CVPR.2016.90
20. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition, 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc 2015 [Online]. Available: http://www.robots.ox.ac.uk/. Accessed 20 Jun. 2023
21. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks. Adv Neural Inf Process Syst. pp. 4468–4476 [Online]. Available: http://arxiv.org/abs/1707.01629. Accessed 20 Jun. 2023
22. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. Proc - 30th IEEE Conf Comput Vis Pattern Recognition CVPR 2017. pp 5987–5995. https://doi.org/10.1109/CVPR.2017.634
23. Zhang H et al (2022) ResNeSt: Split-attention networks. IEEE Comput Soc Conf Comput Vis Pattern Recognit Work 2022:2735–2745. https://doi.org/10.1109/CVPRW56347.2022.00309
24. Kingma DP, Ba JL (2015) Adam: A method for stochastic optimization. 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc [Online]. Available: https://arxiv.org/abs/1412.6980v9. Accessed 26 Jun. 2023

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.