IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2023, accepted 23 February 2023, date of publication 3 March 2023, date of current version 8 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3251988

RESEARCH ARTICLE

A Novel Data Augmentation Method for Improved Visual Crack Detection Using Generative Adversarial Networks

EFSTATHIOS BRANIKAS, PAUL MURRAY, (Member, IEEE), AND GRAEME WEST, (Member, IEEE)

Department of Electronics and Electrical Engineering, University of Strathclyde, G1 1XQ Glasgow, U.K.

Corresponding author: Efstathios Branikas (efstathios.branikas@strath.ac.uk)

ABSTRACT Condition monitoring and inspection are core activities for assessing and evaluating the health of critical infrastructure spanning from road networks to nuclear power stations. Defect detection on visual inspections of such assets is a field that enjoys increasing attention. However, data-based models are prone to a lack of available data depicting cracks of various modalities and present a great data imbalance. This paper introduces a novel data augmentation technique by deploying the CycleGan Generative Adversarial Network (GAN). The proposed model is deployed between different image datasets depicting cracks, with a nuclear application as the main industrial example. The aim of this network is to improve the segmentation accuracy on these datasets using deep convolutional neural networks. The proposed GAN generates realistic images that are challenging to segment and under-represented in the original datasets. Different deep networks are trained with the augmented datasets while introducing no labelling overhead. A comparison is drawn between the performance of the different neural networks on the original data and their augmented counterparts. Extensive experiments suggest that the proposed augmentation method results in superior crack detection in challenging cases across all datasets. This is reflected by the respective increase in the quantitative evaluation metrics.

INDEX TERMS Crack segmentation, generative adversarial networks (GANs), nuclear inspections, data augmentation, image-to-image translation.

I. INTRODUCTION

Health inspection, structural monitoring and fault detection are fields that cover a wide area of applications and meet challenges from different fields like mechanical systems [1], electronics [2] and structural engineering. Of particular interest is the latter case as inspections in structural assets can be performed visually, therefore allowing the use of a broad range of applications and methods.

In the nuclear sector, health inspection processes are of great importance as they allow the condition assessment of critical components. In the UK, the vast majority of nuclear power stations belong to the Advanced Gas-Cooled Reactor (AGR) family and are near the end of their extended lifespan.

Therefore, inspections are carried out regularly to assess the condition of the inspected infrastructure, detect any potential damaged components, and allow the return to operation, provided all the high standards of safety are met. Previous work [3] has focused on developing a piece of software that automatically creates stitched image-montages of the whole inner cylindrical surface of the graphite fuel channels within AGR cores, commonly known as "chanoramas" (channel panoramas). It is therefore imperative that the inspections are conducted efficiently and accurately, to minimize the off-line time of the components under investigation. Assessment of graphite channels within the AGR core is conducted with Remote Visual Inspections (RVI) as the hazardous conditions do not allow direct inspections. The detection and sizing of defects following the data acquisition is conducted manually by human experts and is a time-consuming, tedious, and partially subjective process. Developing assisting methods

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

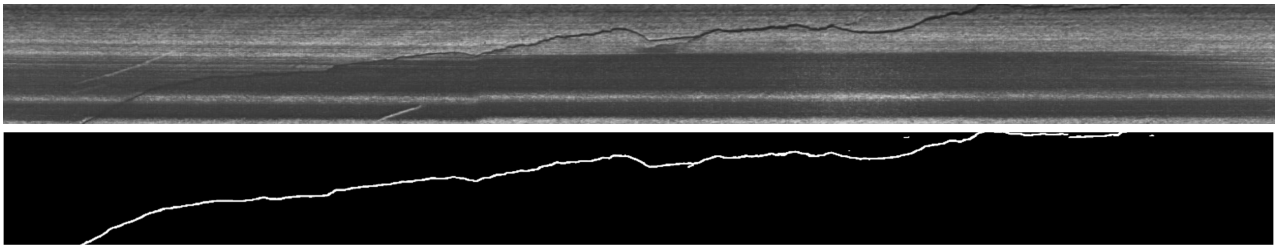


FIGURE 1. An example of a smaller part of an inspection image depicting some of the challenges present in the visual data, like varying contrast and shading in the inspection surface. The respective crack detection map generated by an image segmentation network is also presented.

that automatically detect cracks and defects is a field of active research as it can drastically decrease the data analysis duration and provide the human-in-the-loop with valuable feedback.

Industrial monitoring is lately focusing more on learning-based methods belonging to the field of artificial intelligence and machine learning [4]. These approaches are preferred for their high predictive capabilities and the fact that the models are agnostic to the data, thus requiring minimal modifications. The most widely used methods are Artificial Neural Networks (ANN) as they outperform most models and can be adjusted to perform in a variety of applications. However, the area where ANNs really excel is in image recognition and computer vision tasks; one of the most common tasks in structural health monitoring is crack detection. Visual crack detection is a research field that is increasingly gaining more attention with applications on structural assets like buildings and bridges [5], roads and pavements [6], [7] and even nuclear power stations [8]. The fact that deep learning approaches for real-time crack detection have been developed [9], further encourages the development of similar methods for nuclear power inspections, where the data analysis duration is of great importance.

Crack detection poses many challenges for image recognition algorithms. To perform pixel-level detection, commonly known as *image segmentation*, an annotation process of the defects is required to obtain training pairs between images and the desired output of the algorithm; this process is labor-intensive, time-consuming, and often subject to human bias. A significant challenge in the nuclear case is that cracks tend to occupy proportionally only a small fraction of the overall inspection image or video surface which can be hundreds of Megapixels [10].

An example of this can be seen in Fig. 1. This creates a great imbalance in the data, which leads to a bias in the model to discriminate against classifying a pixel belonging to the positive class. Moreover, inspections are typically long processes, especially in the nuclear field, thus generating great amounts of data. Defects and cracks are not commonly encountered in inspections but are scarcely distributed, or even absent in long inspections. This scarcity of cracks enhances the imbalance of the data even further, requiring specially engineered datasets or modifications to the detection algorithms for correct performance.

To tackle problems like the ones mentioned above, data augmentation methods are developed. For generation of realistic data, Generative Adversarial Networks (GANs) have excelled in visual tasks since their creation [11]. The generative capabilities of GANs include a range of tasks like style transfer and image-to-image translation [12], [13].

GANs have been steadily incorporated as a research tool in the field of fault detection and monitoring. Earlier works include using GANs to create artificial wind turbine fault information data for better fault detection [14], denoising acoustic emission signals for better rail crack detection [15], or even for generating 3D contact-stress distributions on tire-pavement surfaces [16], a novel application of GAN models in the structural inspection field, with many more possibilities for future exploration. Recently, adversarial models have also been used for the task of visual crack detection. In, [17] a GAN model is deployed for direct unsupervised crack detection; the GAN is trained to learn the texture space of the data free of defects and perform detection through a classification of the difference between original and texture images. While useful for classification, this approach does not perform accurate image segmentation because of the lack of annotation. Similarly, in [18] a GAN is used on the output of a convolutional autoencoder to improve the classification accuracy on steel defects, showcasing how generative models can assist health inspection systems.

The authors of [19] propose a framework where a GAN model is trained to reconstruct x-ray images of tires for defect detection. However, this work is based on the fact that the GAN model is not trained on defected images, so it will reconstruct distorted images when the input contains defects, providing less control of the generation. In [20] the authors deploy a GAN-based model for detecting defects and anomalies on solar cell manufacturing. Although a novel approach for such a task, this work doesn't exploit the capabilities of GAN models for generating new samples, which is where GANs excel at. In [21] the authors implement a GAN topology to generate examples of structural adhesive applications for visual quality inspections. Although this approach is similar to ours, the image generation is not based on an image-to-image translation manner, which can create very realistic images, and also does not allow pixel-level detection, without additional annotation costs, which is the main contribution of our proposed approach. Similarly, in [22] the authors create a GAN framework for generating ultrasonic images for non-destructive industrial evaluation, but the evaluation of the generated images takes place manually by human experts, which is costly, and hard to replicate. The CycleGAN model [13] is deployed in [23] to perform road crack segmentation using binary segmentation masks from random objects. Although this innovative approach produces labor-free segmentation results, as the generated labels and the ones used for training are forced through dilation to be skeletons of the cracks themselves, pixel-level correspondence to the cracks cannot be achieved and region-based results are reported instead.

Inspired by the capabilities of GANs on image generation, and their growing presence in the inspection health monitoring research, we present and evaluate a novel framework for nuclear crack data augmentation through domain transfer from different crack datasets. We also assess how our model generalizes on similar applications through deployment on public benchmark data. Our work is most similar with [24], where the authors deploy a GAN to augment pavement crack detection and compare the predictive accuracy between the enhanced and the standard dataset. However, the authors of [24] perform significant manual pre-processing and use a variational autoencoder to extract suitable features for training the GAN. Moreover, the generated images are new, so no ground truth maps are available meaning that image segmentation cannot be performed directly. As an improvement, we instead use CycleGAN [13] to simultaneously transfer crack morphologies between two different domains, maintaining the shape of the source dataset to the style of the target. As both datasets are already annotated, every image that is generated by our GAN is mapped to a respective binary label from the source dataset from which it originated with no additional annotation cost. In this way both datasets are augmented with cracks of different morphologies, adding variance to them at no extra cost. The key advantage of this over the previous approaches using GANs for direct segmentation is that we make use of the well-established generative capabilities of GANs. This is used as an augmentation step to enhance the training sets using existing data that is already labelled. In this way we create examples that are underrepresented and challenging to detect. Moreover, our model is applicable on any segmentation method used for crack detection tasks, to potentially attain superior results. The main contributions of this work are summarized as follows:

1) We deploy CycleGAN, a cyclic-consistency GAN to augment crack detection datasets on highly diverse applications, including nuclear reactor fuel channel surfaces, concrete road, and stone surfaces to acquire realistic examples of cracks that are under-represented in the original datasets.
2) This proposed model is, to the best of our knowledge, the first approach to perform style transfer on structural crack data. In doing so, extra training material is created for the respective datasets without any additional manual annotation as the original images are already annotated. This allows a plug-and-play approach that poses no model restrictions to the subsequent defect detection process.
3) Extensive experiments are carried out to demonstrate the additional data generated by our approach contributes to superior image segmentation results, both on the nuclear industrial case, and the public datasets used for exploring generalization. To this end, various data combinations between real and artificial images are explored.
4) We deploy U-Net as a state-of-the-art model and compare the segmentation results between the augmented data and the original, while providing an ensemble-learning scenario. The findings indicate that, when used for training, the GAN-generated images significantly increase the predictive capabilities of the segmentation across all datasets.

The remainder of the paper is structured as follows. In Section II the basics of generative adversarial networks and the proposed architecture of CycleGAN are described. Section III details the proposed new data augmentation framework, including the implementation and the datasets used, while in Section IV all the experiments carried out are presented in detail. Finally, in Section V all the conclusions of this work are drawn.

II. GENERATIVE ADVERSARIAL NETWORKS

A. ADVERSARIAL LEARNING

The basis of GAN models, since their introduction in [11] has been of an adversarial nature. The key idea is that two different neural networks, the generator (G) and the discriminator (D) are competing against each other. The generator's aim is to capture the distribution of the training data, while the discriminator attempts to distinguish between the real and the generated images.

An equilibrium state is achieved when the generator can produce samples for which the discriminator cannot distinguish between, providing equal probabilities. Fig. 2 provides a graphic representation of a general GAN framework. Both networks are trained together in an adversarial manner through error backpropagation of a common loss function. The generator's input can be an image, or a random

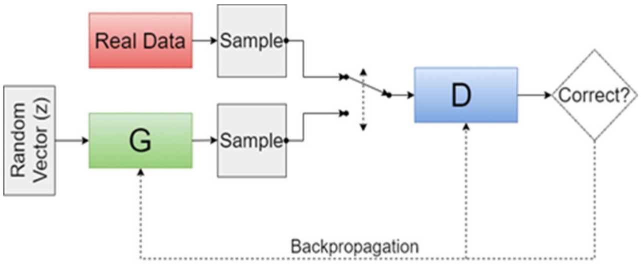


FIGURE 2. A basic generative adversarial network framework. The discriminator alternatively accepts real and generated inputs and aims to classify them accordingly.

vector z that serves as noise from a prior distribution with variables $p_z(z)$.

In this way, the generator's aim is to produce varying samples from the data distribution x through the mapping $G(z)$. The discriminator's output is a scalar value denoting the probability of the input belonging to the real distribution. Both networks then can be updated through the objective function $L_{avd}(G, D)$:

$$\min_G \max_D L_{avd}(G, D) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

Equation (1) is a binary cross entropy loss. Here, the generator attempts to maximize the loss through the term $D(G(z))$. This corresponds to the discriminator's prediction when the input it receives originates from the generator. This is the bottom path starting from the random vector in Fig. 2. The generator aims in maximizing this term which corresponds to the discriminator classifying this as a real example, and being ''tricked'', which increases the error for these predictions. At the same time, the discriminator's aim is to maximize both $\log(D(x))$ and $\log(1 - D(G(z)))$ which correspond to both input paths of the discriminator block in Fig. 2. The maximization of the former term suggests the discriminator detects the real examples, while the latter minimizes the error for detecting the generated examples. Through an alternating optimization and update of the gradients by error backpropagation, the 2 models are trained until convergence.

The expected value in Equation (1) and the following equations correspond to the respective values of the distributions that are denoted. In practice, the values of the losses are calculated across the batch size.

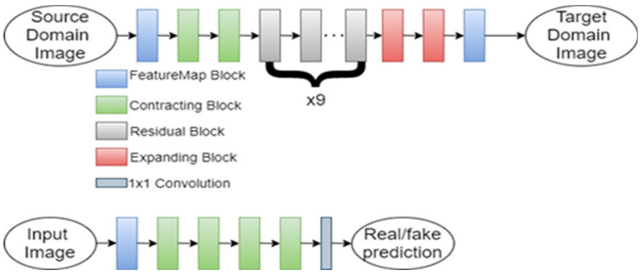


FIGURE 3. The architecture of the Generator (top) and the Discriminator (bottom) networks.

B. CycleGAN MODEL

Since the creation of GANs, numerous pieces of work and different architectures have been developed, serving various tasks, but all have maintained an adversarial training as their key component. CycleGAN is one of the most widely used GAN architectures; it performs image-to-image translation, a general task under which various applications can be fit, ranging from style transfer [25] to image segmentation [23]. In this research study, we also implement the CycleGAN model, as it will be described in the following paragraphs.

The basic CycleGAN architecture can be seen in Fig. 3. The aim of this model is to perform translation of images between two domains, X and Y. This architecture performs a mutual translation in parallel and in an unpaired manner. This imposes no need for pairs of images and their respective counterpart in the second domain, allowing the use of various datasets of different sizes. This mapping takes place using two generators: $G : X \rightarrow Y$ and $F : Y \rightarrow X$. Two discriminators are also introduced, D_X and D_Y to distinguish between real samples $x \in X$ and $y \in Y$ or mapped images on domains X and Y respectively. The overall objective function for training the model consists of three different loss functions: an adversarial loss, similar to that in Eq. 1, a cyclic consistency loss, and an identity loss.

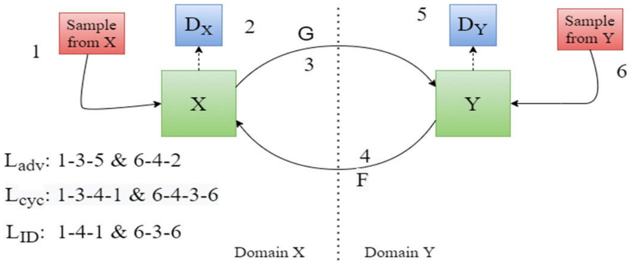


FIGURE 4. The architecture of the CycleGAN model. Numbers correspond to either domain samples, generators, or discriminators for better comprehending the flow of information for different loss functions.

The adversarial loss function tries to match the distribution of the source domain to that of the target through the generated images. For the mapping $Y \rightarrow X$ the loss function can be expressed as:

$$L_{adv}(F, D_x, X, Y) = \mathbb{E}_{x\sim p_{data}(x)}[log(D_X(x))] + \mathbb{E}_{y\sim p_{data}(y)}[\log(1 - D_X(F(y)))] \quad (2)$$

A similar term, $L_{adv}(G, D_Y, X, Y)$ is used for the translation task $X \rightarrow Y$. In theory, this architecture can learn any mapping from one domain to the other. It is therefore desirable to limit these mappings to be reverses of each other and bijections, in order to produce controllable translations between the domains. This can be enforced implicitly by using a cycle consistency loss [13]:

$$L_{cyc}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(G(x) - x)\|_1\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(F(y) - y)\|_1\right] \quad (3)$$

The loss for each mapping aims to minimize the reconstruction error between samples x and y and their respective cyclic mappings to the other domain and back to the original. These losses are represented graphically as paths 1-3-4-1 and 6-4-3-6 in Fig. 3. Finally, an identity loss is introduced:

$$L_{ID}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(x) - x\|_1\right] + [\|G(y) - y\|_1] \quad (4)$$

Equation (4) corresponds to paths 1-4-1 and 6-3-6 in Fig. 3. These terms ensure the generators do not impose unnecessary transformations to the source images that could otherwise be valid mappings. This is implicitly enforced by the loss; if the generators impose additional transformations, other than mapping an image to the target domain, the identity loss will penalize that. Paths 1-4-1 and 6-3-6 essentially feed an image from one domain to the generator that attempts to map it to the same domain. If additional, unintended transformations are applied through the generators, the identity loss will quantify this as the difference between the original and the altered version of the image. With the identity loss, the generators try to negate these additional transformations directly, rather than trying to counter the unwanted effect through the cyclic loss, after they are imposed. The identity loss has proven to limit the possible mappings between domains and maintain the source image composition [13].

The structure of the generator follows that of [12]. Inspired by the Pix2Pix model, we follow a similar approach. A deep autoencoder serves as the generator, where input images pass through a number of encoding blocks and are eventually restored through respective expanding blocks. We also use residual blocks between the contracting and the expanding path as the backbone of the network. All the blocks consist of convolutional layers and non-linear activations; in contrast to traditional architectures used for detection purposes, no pooling operations are used, as they tend to create blurry generated images. Moreover, we substitute instance normalization layers, that are originally used in CycleGAN [13] and in style transfer literature as they allow some droplet artifacts to pass as real through the discriminator. Instead, weight demodulation is selected, inspired by StyleGAN2 [26].

TABLE 1. Components of Network Blocks.

Block	Layers
Feature Map	7x7 convolution + demodulation
Contracting	3x3 convolution + demodulation ReLU activation
Expanding	Bilinear up-sampling 3x3 convolution + demodulation ReLU activation
Residual	3x3 convolution + demodulation ReLU activation 3x3 convolution + demodulation Summation with block input

The discriminator belongs to the patchGAN architecture, similarly to [12]; it classifies images at a patch scale and is convolutionally applied across the surface of the input image, providing a prediction per patch. Fig. 4 depicts the basic structure of the generator and the discriminator, while the respective block architectures are presented on Table 1.

III. DATA AUGMENTATION FRAMEWORK

In this section, the architecture and the functionality of the proposed method as well as a detailed description of the implementation, the datasets used, and the evaluation of the augmentation process are presented.

A. CRACK DETECTION FRAMEWORK

The proposed model is trained and tested using three different datasets. These datasets consist of a small number of images each, while structural cracks have a very limited surface, resulting in datasets with very few examples of cracks available for training. In addition, different modalities of defects are present in the data, leaving even less training data per type of crack. The main application focuses on an industrial case where the dataset has been created using real inspection footage captured during inspections of graphite fuel channels in the UK's fleet of nuclear power stations. Two other publicly available datasets were also used to allow benchmarking and to demonstrate how the proposed approach is able to generalize to other, related applications. These two datasets correspond to concrete pavement [27] and stone cracks [6] respectively. Detailed descriptions for each are presented below.

The main industrial application where we assess our approach is done using a dataset assembled by the authors using video inspection footage of fuel channels that form the reactor cores of the UK's fleet of nuclear power stations. The resultant dataset created from these chanoramas consists of 108 images of varying sizes [8]. The images are highly diverse and depict cracks in a variety of different conditions. Variations include dynamics in lighting, image texture and reflectance, illumination and contrast between the defect and the background. These variations all present challenges for any image processing task that is required to detect the presence of cracks in spite of so many dynamics in the data itself. The resulting original training set on which a segmentation model was trained before the data augmentation consists of 78000 patches.

CrackForest dataset (CFD) [27] is one of the most widely used datasets in the field of crack detection using image processing; it consists of 118 images of road pavements with cracks in different conditions including shadows, spots, and stains. CFD is established as a benchmark dataset for testing the capabilities of crack detection algorithms [27], [6], [7]. The image resolution is 480 by 320 pixels. The cardinality of the original training set for the detection of the cracks is equal to 112000 patches.

Stone331 [6] consists of 331 images of rock surfaces with cracks, as rocks tend to develop cracks on the cutting surfaces. The lighting of the dataset is relatively uniform but different surface textures are present in the data, and the contrast between cracks and the background also varies. The resolution of the stone images is 1024 by 1024 pixels. The size of this training set is approximately equal to 55000 patches.

The process for extracting the training patches for the crack segmentation from the original images is the same as the one described in detail in [8]. For the AGR and the Stone331 dataset, the patch size is 96 pixels per side. For the CFD, as the original images are significantly smaller in terms of resolution, the extracted patches are 64 pixels wide per side.

B. CycleGAN IMPLEMENTATION AND TRAINING DETAILS

The implementation and the training process of the proposed CycleGAN architecture is carried out using Python and the PyTorch deep learning library. To accelerate the training process, the optimization and the training calculations are carried out using GPU instead of CPU devices. The Adam optimizer is selected with a batch size of 5 images for both generators and the optimizer's initial learning rate is set to 0.0001. The batch size was empirically selected for a reasonable trade-off between GPU memory, and speed of convergence. For validating the model, 15% of the training data is used. After an epoch of training is completed, the validation data are used, and the score of the loss functions is used as an evaluation metric. Moreover, some examples are generated per epoch, for visually assessing them manually, as this is the gold standard for evaluating generated images. 50 epochs of training are repeated, unless no convergence of the metric is observed. The final model to use for testing is the one with the lowest loss score, provided the respective generated images are realistic. Methods like cross-validation are not used because of the computational toll of CycleGAN, but the strategy described above is shown to be effective and results in convergence of the trained model.

For the adversarial training of the networks, the discriminators are updated alternatively with their respective generators. Using the error backpropagation method, the gradients of the parameters are updated to minimize the objective loss functions described in Section II. For updating the discriminators, only the adversarial loss is deployed, while the generators' outputs (paths 3 and 4 in Fig. 3) are not updated in order not to hinder the generator's abilities. During the optimization of the Generators, all loss functions are used to update the parameters of the networks, as both the cyclic and the identity loss are applied to the generators for better generative results.

The data require minimum preprocessing in order to train the CycleGAN model. Patches with dimensions 180 by 180 pixels are extracted from each dataset and are directly inserted to the model. The patches were selected as the ones where the predictions contained the most failures in detecting a crack, based in a segmentation model trained in this dataset [8]. A small number of 100 patches with cracks per dataset is found to be adequate for training the model as every batch of patches from one domain is combined with all the examples of the opposite domain within one epoch of training. 10% of the training data is used for validation purposes. As the CycleGAN topology is symmetric, data from both domains are required to have an equal number of input channels. For this reason, the CFD images were converted to grayscale to match the AGR and Stone data. The generated grayscale images were later used for evaluating the augmentation improvement of crack detection on the grayscale version of the dataset. For every dataset, an additional 10% of patches is generated for augmentation and incorporated to the training set.

In Fig. 5 some examples of how the trained generator networks map images between two domains are illustrated. In these examples, the mapping takes place from the CFD to the Stone331 domain. It can be seen that the morphology of the original crack is maintained and transferred, while the style of the image changes to resemble the grain-like surface of the stone, similarly to Fig. 6.

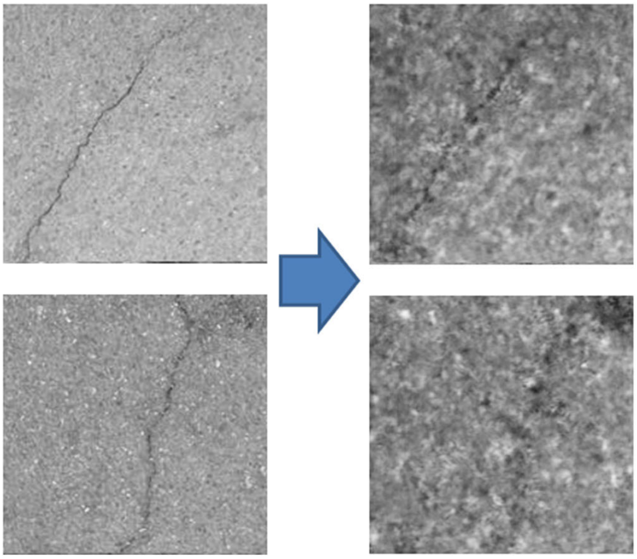


FIGURE 5. Examples of the domain transfer between CFD (source) and Stone331 (target) domain. The input images on the left are mapped to the images on the right.

C. DATA AUGMENTATION EVALUATION PROCESS

The purpose of the proposed domain transfer for augmenting existing datasets is to potentially improve the predictive capabilities of crack detection models by creating more training examples of challenging and underrepresented cases. To evaluate the proposed model, we implement U-Net [28], an image segmentation neural network that is well-established in the field of visual crack detection [7] and with many variations developed recently [6], [8]. U-net is an autoencoder neural network, consisting of blocks of convolutions, non-linear activations and pooling layers. For the interested reader, more details on the network implementation and how it is trained for providing pixel-level crack predictions can be found in [8].

The evaluation is performed by comparing the segmentation results produced by U-Net when applied to the testing set of each dataset between the model trained on the original training set and the augmented counterpart. For the augmented test results, an ensemble of the results of two different U-Net segmentation models is presented; a model trained on the augmented training set, and a model pre-trained on real data and trained subsequently for a small number of epochs on generated data only. This method is selected as experiments show that both models contribute to detecting different aspects of cracks.

The evaluation metrics reported on the testing set of each dataset are the precision, the recall, and the F1 score and are defined as:

$$\text{Precision} = \frac{\mathit{TruePositives}}{\mathit{True\ Positives} + \mathit{False\ Positives}} \tag{5}$$

$$\text{Recall} = \frac{\mathit{True\ Positives}}{\mathit{True\ Positives} + \mathit{False\ Negatives}} \tag{6}$$

$$\text{F1 score} = \frac{2 * \mathit{precision} * \mathit{recall}}{\mathit{precision} + \mathit{recall}} \tag{7}$$

Following the same approach as in [7], [22], and [8], we do not consider false positives that are within a 5-pixel radius from true positive predictions as errors. In this way, the model is not penalized for "thick" predictions. We also allow a radius for not penalizing false negative predictions that are in the close vicinity of true positive predictions, similarly to [8]. Within this radius, pixels that are false negatives, are treated as true negatives.

This relaxation is introduced to counter mismatches present in the label maps of the data, as a result of human bias. In this way, the reported metrics reflect in a fairer way on the model's capability to detect a crack, rather than penalizing a thinner prediction or errors in ground truth maps.

IV. EXPERIMENTAL RESULTS

The comparative results of the crack detection process between the U-net model trained on the original data and the augmented versions of the data using our new training approach is summarized in Tables 2 – IV for all the respective datasets. Column "R" corresponds to the different values of the relaxation radius introduced (as described in Section III) for false negative predictions and refers to the pixel value of the radius. Experiments across all different datasets suggest a clear improvement in crack detection when generated data are included in the training space of the algorithm. This is reflected by the recall metric which corresponds to the ratio of cracks correctly detected.

It is worth mentioning that the precision metric remains unchanged for different values of R, as the relaxation associated with R handles the affected pixels as true negatives, which does not affect the precision metric, as can be seen in Equation (5).

In the case of the AGR data, the improvement attributed to the augmentation is witnessed when a few pixels of relaxation radius are deployed. This dataset was not annotated using specialized tools or human experts; therefore, it benefits most from the relaxation. For a task like crack detection, the most important task is not to miss any cracks when they are present, because of the challenges imposed by the class imbalance and the scarcity of cracks. Fig. 5c depicts the most challenging defect example where the original model ignores the visual crack almost completely. The model trained on artificial data significantly improves the prediction, setting a new state-of-the-art standard for this dataset on these highly diverse images. Furthermore, the augmentation with CFD improves the precision of the model as well, which can be seen in Table 2.

TABLE 2. Crack Detection Results on AGR Dataset.

	Original		Augmented (CFD)		Augmented (Stone)	
	(Prec = 0.988)		(Prec = **0.993**)		(Prec = 0.962)	
R	Rec.	F1	Rec.	F1	**Rec.**	**F1**
0	0.807	0.888	0.799	0.886	0.807	0.877
1	0.873	0.927	0.874	0.930	0.876	0.917
2	0.910	0.948	0.919	0.955	0.918	0.940
3	0.927	0.956	0.939	0.965	0.940	0.951
4	0.936	0.961	0.950	0.971	**0.951**	**0.956**

TABLE 3. Crack Detection Results on CFD Dataset.

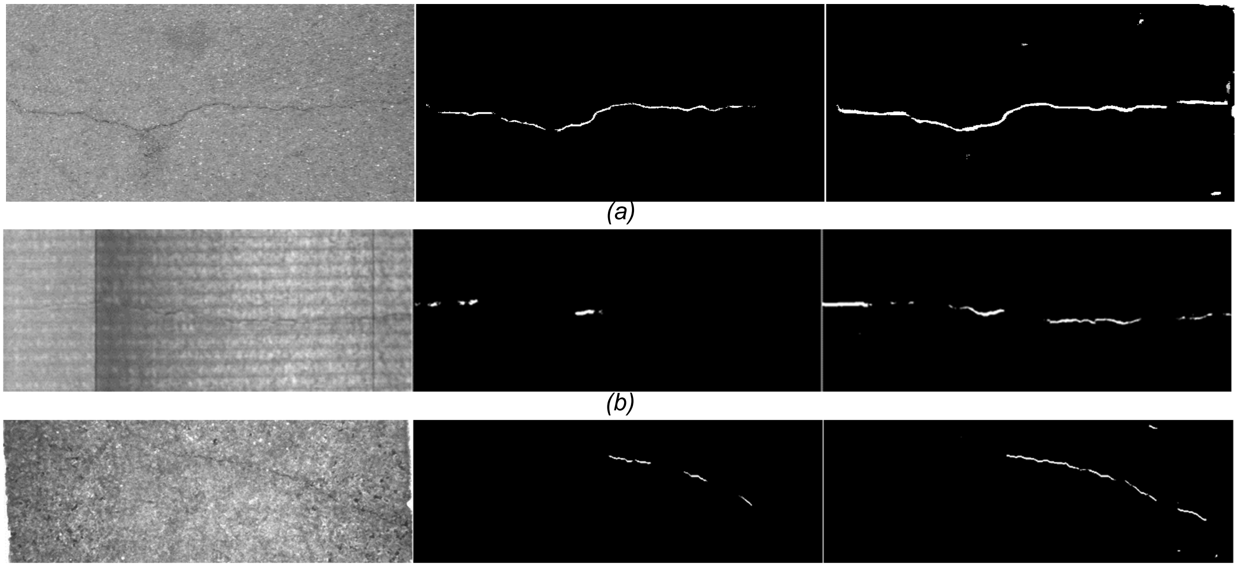
	Original		Augmented (AGR)		Augmented (Stone)	
	(Prec = **0.996**)		(Prec = 0.942)		(Prec = 0.990)	
R	Rec.	F1	**Rec.**	F1	Rec.	**F1**
0	0.705	0.825	0.857	0.897	0.798	0.884
1	0.845	0.915	0.923	0.933	0.906	0.946
2	0.880	0.935	0.941	0.941	0.931	0.960
3	0.896	0.943	0.949	0.946	0.942	0.965
4	0.907	0.949	**0.955**	0.948	0.949	**0.969**

TABLE 4. Crack Detection Results on Stone331 Dataset.

	Original		Augmented (AGR)		Augmented (CFD)	
	(Prec = 0.979)		(Prec = **0.985**)		(Prec = 0.952)	
R	Rec.	F1	Rec.	F1	**Rec.**	**F1**
0	0.676	0.800	0.692	0.813	0.741	0.833
1	0.732	0.837	0.753	0.853	0.793	0.865
2	0.743	0.845	0.764	0.860	0.805	0.873
3	0.750	0.849	0.770	0.864	0.812	0.876
4	0.756	0.853	0.776	0.868	**0.817**	**0.880**

Based on the results presented on Table 3, augmenting CFD with both the AGR and the Stone331 dataset greatly improves the crack detection efficiency, reflected on a significant increase in the recall metric even when no relaxation is applied. The AGR case provides a higher recall at the cost of precision (evident also in Fig. 5a) while the Stone331 provides a smaller improvement in recall, almost without compromising the precision metric; in both cases the proposed method significantly increases the detection capabilities of the segmentation model. Although the grayscale version of this dataset is used, the augmented results are on par with state-of-the-art methods used on the original CFD. However, a direct comparison with other detection algorithms would not be fair, as the proposed U-Net model is trained on different data. The example presented in Fig. 5 clearly shows that the proposed model detects challenging cases of low contrast and faint cracks.

Results on Stone331 dataset also depict a clear improvement in crack detection; the respective recall values on Table 4 for both AGR and CFD augmentation are higher than that of the original training set. Although augmenting this dataset with AGR data provides a higher precision as well, as the CFD results increase the recall by a greater margin, the F1 score is higher in the case of the CFD augmentation, even though there is a small decrease in the precision metric. In both cases, the crack detection rate is superior, and this is



(a)

(b)

(c)

FIGURE 6. Testing examples between crack detection results obtained from training U-Net with only real data (middle) and the respective augmented datasets (right). Images correspond to CFD (a), Stone331 (b) and one of the most challenging examples of the AGR dataset (c).

also evident in Fig. 5b where representative improvements in detecting these defects are presented.

V. CONCLUSION

Structural health inspection monitoring and defect detection is a field that is suffering from data imbalances when quantifying this problem for learning-based methods while acquiring fully annotated datasets is extremely time consuming. It would be desirable to tackle these challenges in a manner that can both generalize across different data modalities and require minimal manual work. Therefore, in this paper an approach to this end is presented, by modifying and implementing a well-established generative adversarial network for a cyclic image-to-image translation to perform domain transfer in two domains. In this way an augmentation is achieved on both domains and in the case where an existing annotation is available for only one dataset, the new generated data on the other domain can share this annotation, creating an artificial annotated dataset. Through extensive experiments conducted in three different test cases, it is shown that the proposed augmentation framework alleviates the challenges mentioned above and improves the crack segmentation process, providing state-of-the-art results in some cases.

The main contribution of this research lies on the fact that already annotated datasets are used for the augmentation process. In this way, through transforming data, as illustrated in Fig. 5, new examples are generated that maintain the structural information of the defects. As the binary maps from the original domains are reusable for the new cases, the additional data can be used directly as extra training material without any annotation cost. This is the biggest difference with other similar approaches, as the generated data can be used directly. To our best knowledge, this is the first time GAN-based data augmentation is exploited for structural crack segmentation, and the generated images are already annotated. The evaluation of the augmented data shows that indeed the proposed framework improves the segmentation performance.

As one specific segmentation neural network is deployed to test the effectiveness of the proposed augmentation framework, in future work it could be beneficial to further investigate how the GAN topology is associated with the detection results of other segmentation algorithms. In this way a comparison between U-Net and other state-of-the-art methods could further cement the proposed model's effectiveness. It is outside the scope of this research paper to deploy different detection models, as our main aim is to establish the framework for generating the images using CycleGAN, and compare the detection results with and without the augmentation data. Comparing with other approaches is not feasible, as performance on the augmented data would be compared against other algorithms on the original data. In this paper, U-Net model was selected for its wide success in many image segmentation methods [28], with crack detection being one of them [7], [8]. In all the experiments, including the original datasets, the precision was very high, which can be attributed to the nature of this challenge and the inherent class imbalance. The purpose of the augmentation was to improve the crack detection rate, and this is achieved in all the cases, resulting both in higher recall metrics reported, and in many cases an overall better F1 and precision score.

ACKNOWLEDGMENT

The authors would like to thank EDF Energy for kindly providing the data corresponding to the Advanced Gas-Cooled Reactor dataset. The image generation results were obtained using the ARCHIE-WeSt High-Performance Computer (www.archie-west.ac.uk) based at the University of Strathclyde.

REFERENCES

[1] X. Gong and W. Qiao, "Current-based mechanical fault detection for direct-drive wind turbines via synchronous sampling and impulse detection," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1693–1702, Mar. 2015, doi: 10.1109/TIE.2014.2363440.

[2] A. Eskandari, J. Milimonfared, and M. Aghaei, "Fault detection and classification for photovoltaic systems based on hierarchical classification and machine learning technique," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12750–12759, Dec. 2021, doi: 10.1109/TIE.2020.3047066.

[3] G. West, P. Murray, S. Marshall, and S. McArthur, "Improved visual inspection of advanced gas-cooled reactor fuel channels," *Int. J. Progn. Health Manag.*, vol. 6, no. 3, pp. 1–11, Nov. 2020, doi: 10.36001/ijphm.2015.v6i3.2269.

[4] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014, doi: 10.1109/TIE.2014.2301773.

[5] S. O. Sajedi and X. Liang, "A convolutional cost-sensitive crack localization algorithm for automated and reliable RC bridge inspection," in *Risk-Based Bridge Engineering*. Boca Raton, FL, USA: CRC Press, 2019, pp. 229–235, doi: 10.1201/9780367815646-19.

[6] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning hierarchical convolutional features for crack detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019, doi: 10.1109/TIP.2018.2878966.

[7] M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, and G. Morison, "A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2120–2124, doi: 10.23919/EUSIPCO.2018.8553280.

[8] E. Branikas, P. Murray, and G. West, "An innovative crack detection algorithm to support automated inspection of nuclear reactor cores," in *Proc. 12th Nucl. Plant Instrum., Control Human-Mach. Interface Technol.*, 2021, pp. 1–10.

[9] W. Choi and Y.-J. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020, doi: 10.1109/TIE.2019.2945265.

[10] P. Murray, G. West, S. Marshall, and S. McArthur, "Automated in-core image generation from video to aid visual inspection of nuclear power plant cores," *Nucl. Eng. Des.*, vol. 300, pp. 57–66, Apr. 2016, doi: 10.1016/j.nucengdes.2015.11.037.

[11] I. Goodfellow, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[12] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, and B. A. Research, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1125–1134.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[14] J. Liu, F. Qu, X. Hong, and H. Zhang, "A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3877–3888, Jul. 2019, doi: 10.1109/TII.2018.2885365.

[15] K. Wang, X. Zhang, Q. Hao, Y. Wang, and Y. Shen, "Application of improved least-square generative adversarial networks for rail crack detection by AE technique," *Neurocomputing*, vol. 332, pp. 236–248, Mar. 2019, doi: 10.1016/j.neucom.2018.12.057.

[16] X. Liu, A. Jayme, and I. L. Al-Qadi, "ContactGAN development—Prediction of tire-pavement contact stresses using a generative and transfer learning model," *Int. J. Pavement Eng.*, pp. 1–11, Oct. 2022. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10298436.2022.2138876, doi: 10.1080/10298436.2022.2138876.

[17] G. Hu, J. Huang, Q. Wang, J. Li, Z. Xu, and X. Huang, "Unsupervised fabric defect detection based on a deep convolutional generative adversarial network," *Textile Res. J.*, vol. 90, nos. 3–4, pp. 247–270, Feb. 2020, doi: 10.1177/0040517519862880.

[18] H. Di, X. Ke, Z. Peng, and Z. Dongdong, "Surface defect classification of steels with a new semi-supervised learning method," *Opt. Lasers Eng.*, vol. 117, pp. 40–48, Jun. 2019, doi: 10.1016/j.optlaseng.2019.01.011.

[19] Y. Wang, Y. Zhang, L. Zheng, L. Yin, J. Chen, and J. Lu, "Unsupervised learning with generative adversarial network for automatic tire defect detection from X-ray images," *Sensors*, vol. 21, no. 20, p. 6773, Oct. 2021, doi: 10.3390/s21206773.

[20] J. Balzategui, L. Eciolaza, and D. Maestro-Watson, "Anomaly detection and automatic labeling for solar cell quality inspection based on generative adversarial network," *Sensors*, vol. 21, no. 13, p. 4361, Jun. 2021, doi: 10.3390/s21134361.

[21] R. S. Peres, M. Azevedo, S. O. Araujo, M. Guedes, F. Miranda, and J. Barata, "Generative adversarial networks for data augmentation in structural adhesive inspection," *Appl. Sci.*, vol. 11, no. 7, p. 3086, Mar. 2021, doi: 10.3390/app11073086.

[22] L. Posilovic, D. Medak, M. Subasic, M. Budimir, and S. Loncaric, "Generating ultrasonic images indistinguishable from real images using generative adversarial networks," *Ultrasonics*, vol. 119, Feb. 2022, Art. no. 106610, doi: 10.1016/j.ultras.2021.106610.

[23] K. Zhang, Y. Zhang, and H. D. Cheng, "CrackGAN: Pavement crack detection using partially accurate ground truths based on generative adversarial learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1306–1319, Feb. 2021, doi: 10.1109/TITS.2020.2990703.

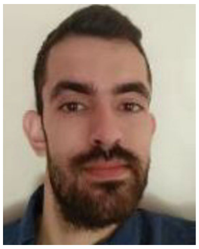
[24] L. Pei, Z. Sun, L. Xiao, W. Li, J. Sun, and H. Zhang, "Virtual generation of pavement crack images based on improved deep convolutional generative adversarial network," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104376, doi: 10.1016/j.engappai.2021.104376.

[25] S. Palsson, E. Agustsson, R. Timofte, and L. Van Gool, "Generative adversarial style transfer networks for face aging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2084–2092.

[26] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 8110–8119.

[27] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016, doi: 10.1109/TITS.2016.2552248.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.



EFSTATHIOS BRANIKAS received the M.Eng. degree in electrical and computer engineering from the University of Patras, Patras, Greece, in 2019. He is currently pursuing the Ph.D. degree with the University of Strathclyde, Glasgow, U.K.

His research interests include computer vision, image segmentation, deep and learning, and causal machine learning.

PAUL MURRAY (Member, IEEE) received the M.Eng. and Ph.D. degrees in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 2008 and 2012, respectively.

He is currently a Reader with the University of Strathclyde. His research interests include image processing, hyperspectral imaging and analysis, feature extraction, and machine learning

GRAEME WEST (Member, IEEE) received the B.Eng. and Ph.D. degrees in electronic and electrical engineering from the University of Strathclyde, Glasgow, U.K., in 1998 and 2002, respectively.

He is currently a Reader with the University of Strathclyde. His research interests include intelligent systems, artificial intelligence, nuclear power generation instrumentation, inspection monitoring, and machine learning.