Contents

[2. Abstract 2](#_Toc114152563)

[3. Intro 2](#_Toc114152564)

[4. Methodology and results 4](#_Toc114152565)

[4.1. Data generation: 4](#_Toc114152566)

[4.1.1. Experimental data collection 4](#_Toc114152567)

[4.1.2. CIVA/simulated data collection 5](#_Toc114152568)

[4.1.1. Signal processing and image generation 5](#_Toc114152569)

[4.2. CNN classification evaluation method 6](#_Toc114152570)

[4.2.1. How we used this for comparison 6](#_Toc114152571)

[4.2.1. Hyperparameter optimization from experimental data 7](#_Toc114152572)

[4.3. Classification results for experimental and direct simulated data: 8](#_Toc114152573)

[4.3.1. Experimental results: 8](#_Toc114152574)

[4.3.2. Simulated results: 8](#_Toc114152575)

[4.4. Approaches for noise generation: 9](#_Toc114152576)

[4.4.1. UT CycleGAN 9](#_Toc114152577)

[4.4.2. Real noise 13](#_Toc114152578)

[4.4.3. Simulated C scan noise 13](#_Toc114152579)

[4.4.4. Simulated A scan noise 15](#_Toc114152580)

[5. Discussion and comparison 18](#_Toc114152581)

[5.1. Comparison of performance 18](#_Toc114152582)

[5.2. Model interpretability with Grad-CAM 19](#_Toc114152583)

[5.3. Discuss results 19](#_Toc114152584)

[6. Conclusion 20](#_Toc114152585)

[7. Acknowledgment 22](#_Toc114152586)

[8. References 22](#_Toc114152587)

Synthetic data generation approaches including GANs for domain adaption of defect classification of Non-destructive ultrasonic testing.

# Abstract

*Abstract—This work provides a solution to the challenge of low training data volumes in Non-Destructive Ultrasonic Testing of carbon fibre reinforced polymer composites known for their high structural ultrasonic noise. We used the strengths of CycleGAN, an unpaired image-to-image translation network, to learn the mapping from physics-based simulations of defects to experimental results. The learnt mapping was then applied to a simulated dataset to closer replicate real experimental data. CycleGAN was able to effectively learn the mapping between simulated and experimental data and a resulting synthetic dataset was produced. By using a Convolutional Neural Network, it was demonstrated that the (non CycleGAN) simulated dataset was too dissimilar to experimental data for direct training of an experimental classifier – producing a classification F1 score of 0. However, when trained on the new synthetic dataset, which was augmented using the GAN, the classifier demonstrated a significant improvement in classification performance on experimental data, with a classification F1 score of 0.88. This showed that the CycleGAN had successfully mapped the simulated data closer to the experimental data, allowing simulated data to be used much more effectively in training.*

# Intro

Composites such as Carbon Fibre Reinforced Polymer (CFRP) are constructed by layering multiple carbon ply layers which are cured after the addition of a polymer. These composites are widely used in aerospace and other industries as they offer superior corrosion resistance, specific strength and stiffness, and their anisotropic nature can be mapped to correspond with structural load requirements [1]–[9]. Composites are susceptible to defects created during manufacturing [1], [2], [4], [7], [8], [10], [11]. The detection, quantification and characterization of these defects is required to assess the quality of aerospace components. These defects most commonly include delamination’s, cracks, foreign object inclusions, fibre distortions (or marcels), and porosity [6], [11]. With an increasing amount of composites becoming safety critical parts, the need for effective testing is of upmost importance [4].

Non-Destructive Testing (NDT) encompasses a range of techniques used to inspect components without causing damage. Some of the most common methods are Radiography, Thermography, Electromagnetic methods, and Ultrasound.

Ultrasonic Testing (UT) has been widely adopted and standardized for testing in the aerospace industry due to its ease of implementation and ability to detect a wide variety of defects [2], [6], [9], [10]. Ultrasonic inspection in NDT works in a similar way to medical Ultrasonography, where sound waves are excited on the surface of a component and the reflections from internal wave propagation can give useful information about the volume of the component. Commonly, phased arrays are used to generate the initial sound wave. This has the advantage of allowing images to be produced by combining multiple distinct point sources of energy [8]. By combining linear phased array probes (Figure 1) with mechanized scanning, UT can produce complete 3-dimensional images of components by stacking multiple individual depth wise images (B-scans) together at known positions. Most often the data is visualized as 2-dimensional B-scans or amplitude C-scans; where the maximum response from a depth gating produces a section view across the component (examples can be seen in Figure 3) [12].

|  |
| --- |
| *Graphical user interface, application  Description automatically generated* |

Figure : Demonstration of how individual probe elements can make up a linear phased array which can produce B-scan and C-scan images.

The use of robotics in NDT has created the ability to automate large scale inspection processes efficiently [13]. However, despite drastic reduction in scan time seen by the introduction of robotic scanning, the interpretation of the results in industry remains a challenging and time intensive task that requires a highly trained and qualified operator [9], [14]–[18]. The requirement for this operator introduces two key issues: a lack of time efficiency and the introduction of human error [16]. Simple automation of results can be seen in the mass production of parts with precisely known geometries. However, this is often based on feature extraction/engineering and is unable to deal with complex variations, for example changes in manufacturing or geometry [18]. Therefore, if a Deep Learning (DL) approach could be created to automate the interpretation of complex results and work alongside the robotic inspection, significant benefits could be seen in the efficient inspection of large components, allowing for further use of UT in aerospace and other industries. With DL being identified as a requirement to transition from low to high levels of industrial automation [18].

However, despite the clear opportunity, Machine Learning has seen limited uptake in UT, particularly for composite components, which present a more challenging case with additional structural noise compared to homogeneous materials. A clear barrier to uptake is the lack of training data [18]. Modern manufacturing processes aim to reduce the production of defects, meaning large volumes of real defect responses are simply not available. Especially ones that represent the full distribution of defect classes and wide variability within these classes that are present from variable manufacturing conditions. Most commonly, previous works aim to experimentally increase their datasets using manufactured defects [5], [19], [20]. However, whilst these can prove research concepts, they are unlikely to give responses that accurately represent real-world responses especially not at the same variability seen within real defects. Other authors have demonstrated success using simulated data developed using Finite Element Analysis (FEA) software to model defects and ray-based models to create Plane Wave Capture, which uses a physics-based understanding of the wave propagation to produce accurate responses based on bulk material properties [21]. However, this is typically done for homogenous steel samples which have very low attenuation and noise, and have less modelling complexity compared to composites, which are acoustically inhomogeneous and produce large amounts of attenuation and noise. Furthermore, this noise is often produced structurally from the internal ply interfaces of the material and is not random. Therefore, merely adding randomly distributed noise may give unrealistic images or obscure defect responses. Most modern simulation software is unable to account for these interactions as materials are modelled using bulk properties and not done at the individual ply level. As an alternative to full FEA software, semi-analytical physics based software has been shown to produce experimentally accurate defect responses [22], [23]. This software is much less computationally expensive than full FEA. Furthermore this software can be used for simulating composite responses based on bulk material properties [24].

Synthetic datasets are widely used in ML to augment small training datasets [25]and they offer a potential solution to the lack of defect data in UT. This work looks at different methods of generating synthetic datasets from simulated data for composite UT. These synthetic data generation methods are comparatively evaluated on their experimental classification performance when used for training a Convolutional Neural Network (CNN). One of these methods is a GAN based approach. GANs have seen success in generating and augmenting training data [25]–[28]. They are often used to augment the distribution of a particular target case, relying on the variability within the GAN to provide a greater variability in training examples. The specific GAN used in this work is CycleGAN. CycleGAN is a conditional GAN which has demonstrated good results in unpaired image-to-image translation tasks [29]. Our GAN approach aims to combine the use of NDT data from physics-based simulations with GAN augmentation to create a dataset based upon physically accurate defect responses that better resemble experimental data. Our approach uses a modified CycleGAN architecture to learn the mapping from simulated data to experimental data. Our modifications help to encourage accurate defect signal reproduction whilst allowing for the addition of experimental noise. With this approach, large quantities of highly varied simulated defects can be produced, and using the GANs mapping, produce large quantities of experimentally representative synthetic data. The overall goal of this work is to identify the best methods for generating synthetic datasets in UT of composites to help unlock the potential of DL in NDT applications.

# Methodology and results

## Data generation:

### Experimental data collection

Three 8.6mm thick composite samples were provided by Spirit AeroSystems. In one sample, 15 Flat-Bottom Holes were drilled form the backside to simulate defects. This will be referred to as the test sample. The defects were 3.0, 6.0 and 9.0mm in diameter, with each individual defect size drilled to depths of 1.5, 3.0, 4.5, 6.0, 7.5mm from the front surface. In another sample, known as the train sample, 25 Flat-Bottom Holes were drilled at the same depths but with additional defect sizes of 4.0 and 7.0mm. The other sample, known as the clean sample, was kept defect-free for generating defect-free images. Flat-Bottom Holes are commonly used in UT to simulate defects [30], in addition to this their consistent geometry makes them simple to simulate. It was important that whatever defects were manufactured could easily be simulated so that we had the most direct comparison between simulated and real data, as we were solely looking at the ability of simulated data to train for experimental classification and not the differences between real defects and manufactured defects. The composite samples were all manufactured to the BAPS 260 specification using a Resin Transfer Infusion Process, made using non-crimp fabric and Cycom 890 resin. The ultrasonic data was collected by linear phased array scanning using a 64-element 5MHz ultrasonic roller probe, with 100V and gain of 22.5dB, which was robotically controlled by a KUKA KR 90 (Figure 2). The robotically controlled scanning allowed for the concatenation of B-scans to produce C scan images. To ensure the acoustic wave energy was consistently transferred into the sample at different scanning positions, a Force-Torque sensor was used to maintain a constant 35N scanning pressure, and water was used as an acoustic couplant due to its closely matched acoustic properties to the rubber of the roller probe tyre. This is a similar acquisition setup to what is used in industry and has been used to collect data on large composite aerospace components [31].

|  |
| --- |
| *A picture containing indoor, equipment  Description automatically generated* |

Figure 2: Experimental setup of KUKA KR90 and ultrasonic roller probe used for experimental data acquisition.

### CIVA/simulated data collection

A simulated dataset of the experimental test sample previously discussed was constructed using a semi-analytical physics-based commercial NDT simulation software. As the software is semi-analytical it allowed for simulations to be completed with much less computational cost than common Finite Element Analysis (FEA) methods. As the focus of this work is the opportunity to produce large datasets for UT, this is a significant benefit of a semi-analytical software and makes the application of complex FEA simulations untenable. However, it is important that the software whilst being efficient can also produce realistic simulation responses. The simulation software which is widely used for commercial UT simulation work is physics-based and has been experimentally validated for UT [22], [23]. Therefore, we could be confident that the wave propagation was accurate and was effective at modeling complex defect responses. In addition, the simulated defect dimensions and positions were controlled, allowing us to duplicate the exact experimental setup. This also allows for efficient, complete annotations of the dataset to be generated at the point of simulation, which opens further opportunities beyond classification, such as segmentation etc. A downside of using a semi-analytical software and not FEA is that the software is unable to model each distinct composite layer response. This is a significant downside and likely explains the significant difference between experimental and simulated data. For our model creation the individual layers were still constructed but were only used to give the bulk material properties. The model was constructed to match the experimental sample as closely as possible with 0, 45, -45, and 90 degree alternating ply layers. The fiber volume was also set to X to give the density which best matched the experimental sample value of 1440kg/m3. A parametric study simulation was setup which used the composite bulk properties previously calculated and varied the diameter and depth of defects. The study matched the experimental setup with 3.0, 6.0 and 9.0mm defects at depths of 1.5, 3.0, 4.5, 6.0, and 7.5mm from the surface.

### Signal processing and image dataset generation

The image resolution was physically limited by the number of array elements to 64 pixels in the array dimension, the second dimension was matched to this by selecting the corresponding 64 B-scans to produce square images. The distance was 0.8mm between elements and the robotic scanning was controlled to give 0.8mm B-scan offset so that the images produce square pixels, which kept the dimensions consistent between the physical component and the ultrasonic data. Since ultrasound values are just amplitude levels of voltage response, the images were kept in grayscale as any colors did not have any physical significance.

Both the experimental and simulated data collected were in the form of radio frequency time traces. Signal processing steps were taken to create amplitude C scan images from the two sources of data. Firstly, the time traces were zero centered and had a Hilbert transform applied. This is standard signal processing for image generation of time series ultrasonic data. The experimental and simulated datasets were then normalized between 0 and 1 by dividing by their respective max values. The normalization allowed for direct comparison of the different datasets.

Once the data was normalized, the data was truncated to remove the front and back wall echoes. Then the max amplitudes were taken at varying depths of 5 samples to produce C scans. From these C scans, the images which represented a defect response were collected. For the experimental samples we also collected the defect free C scan images. This produced 334 defective images from the experimental train sample, 148 defective images from the experimental test sample and 640 defect free images from the clean sample. This was split into 334 clean train images and the rest were used for testing. From the simulated dataset, 154 defective images were produced. Figure 3 shows how the simulated responses were significantly different from the experimental data. The simulated responses were far cleaner than the experimental responses and lacked the structured noise that is typically seen in experimental scans from the composite ply interactions.

|  |  |  |
| --- | --- | --- |
| Data source | Dataset | Number of images |
| Experimental test sample  (15 holes) | Defective test | 148 |
| Experimental train sample  (20 holes) | Defective train | 334 |
| Experimental clean sample | Clean test | 148 |
| Clean train | 334 |
| Simulated experimental train sample  (15 holes) | Simulated defective | 154 |

Table : Summary of the datasets produced

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) | *Chart  Description automatically generated with medium confidence* |  | b) | *Chart  Description automatically generated with low confidence* |

Figure : example of simulated (a) and experimental (b) C scan responses.

## CNN classification evaluation method

### How we used this for comparison

The aim of this work is to evaluate different methods of modifying simulated data to make them more effective at training Deep Learning models for experimental classification tasks. It is therefore important that we evaluate our synthetic datasets with respect to a classification metric.A Convolutional Neural Network (CNN) was used to evaluate and compare the classification performance of different synthetic and experimental datasets.

Since the focus of this work was to compare synthetic datasets and not on optimal classification accuracy, the CNN was kept constant for each dataset. Whilst we wanted to keep the CNN lightweight to reduce the computational cost of testing each synthetic dataset, it was also important that the CNN had adequate complexity to learn the task. To make sure the CNN had enough complexity to represent the solution space we used a genetic algorithm for hyperparameter optimisation (HPO) of a CNN when trained on experimental data.

As we are working with small datasets, there is a degree of variability in our classification results. To negate this, when training the classifier, we re-trained the CNN for each synthetic dataset with a fresh initialisation 100 times and the average results were taken. Each CNN was evaluated on the same experimental dataset.

We constructed confusion matrixes and calculated precision, recall and F1 scores, which are presented in the following equations:

Precision = TP / (TP+FP)

Recall = TP / (TP+FN)

F1 = (2\*Precision\*Recall) / (Precision+Recall)

Where TP is true positive, FP is false positive, and FN is false negative, with positives being the presence of a defect. Each result was individually averaged using a simple mean across the 100 training cycles.

### Hyperparameter optimization from experimental data

We used a genetic algorithm to perform HPO on our experimental train dataset to determine the parameters for the CNN. The CNN was kept simple to save on computational resource. The model had at least 1 convolutional layer. Each convolutional layer had a fixed kernel size of 3 and used ReLU activation followed by max pooling with a kernel size of 2. The number of convolutional layers was parameterised with the number of filters given by a constant out-channel ratio and the number of out channels from the previous layer. The out-channel ratio was also parameterised. The network always had at least one fully connected layer, from the flattened layer to the single output node, with a sigmoid activation function for binary classification. There were a variable number of fully connected layers. Each hidden fully connected layer used ReLU activation. The number of nodes on each hidden layer were equally distributed by dividing the number of nodes in the flattened layer by the total number of layers and removing this from the previous hidden layer each time. The network HPO also included batch size, early stop, learning rate, momentum, and number of epochs. The values for the HPO variables are given in Table 2. Figure 4 shows an example of the network with three convolutional layers, 2 hidden layers, and an out filter ratio of 2.

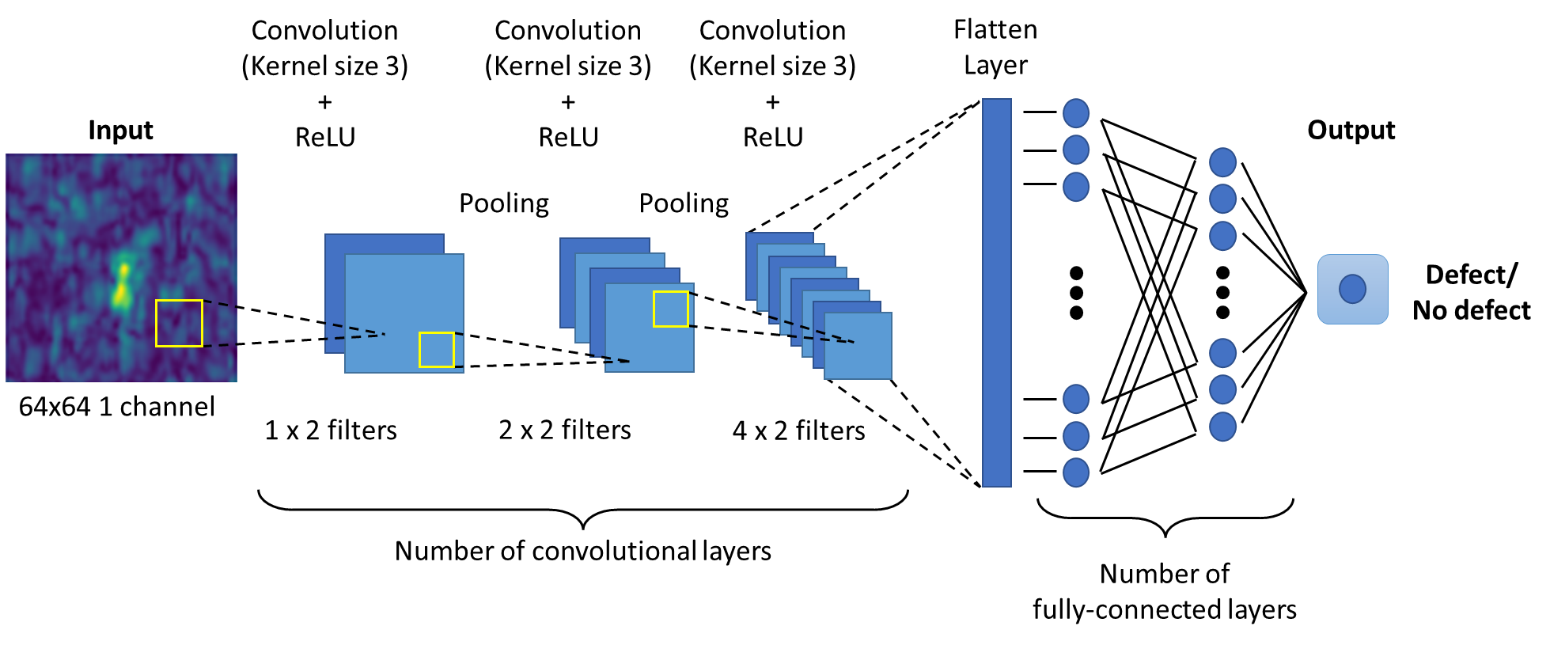


Figure : CNN architecture example with a convolutional channel ratio of 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Variable Parameter | Range | | Number of fully connected layers | 1 - 6 | | Number of convolutional layers | 1 - 6 | | Channel ratio for convolutional layer filters | 1 - 3 | | Batch size | 16, 32, 64, 128, 256 | | Early stop | 0 - 5 | | Learning rate | 0.00001 – 0.5 (log scale) | | Momentum | 0 - 1 | | Number of epochs | 100 - 500 | |

Table : HPO variables and their range of values.

The HPO was performed using the experimental test dataset, made up of 334 defect images and the same number of defect free images from the clean test dataset. The algorithm first performed 128 random mutations. From this a genetic algorithm was used for a total of 512 mutations. For each mutation, the average F1 score over 10 iterations was used as the evaluation metric. The dataset was randomly subsampled for each iteration with 80% of the dataset used for training and 20% used for testing. The optimum final network had an average F1 score of 0.978. The hyperparameters are outlined in Table 2. The network was implemented using the Pytorch framework.

|  |  |
| --- | --- |
| Variable Parameter | Range |
| Number of fully connected layers | 1 |
| Number of convolutional layers | 3 |
| Channel ratio for convolutional layer filters | 3 |
| Batch size | 16 |
| Early stop | 1 |
| Learning rate | 0.014 |
| Momentum | 0.176 |
| Number of epochs | 264 |

Table :Optimized hyperparameters used for CNN.

## Classification results for experimental and direct simulated data:

### Experimental results:

For comparison to synthetic datasets, we trained a model on the experimental test dataset and the same number of clean images sampled from the clean test dataset with a train, test split of 80% and 20% respectively. After 100 training iterations, the model gave average accuracy of 89.8%, with average F1, precision and recall scores of 0.887, 0.974 and 0.826 respectively. The average confusion matrix for the experimentally trained model is given in Table 2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 29.95 | 0.98 | | **No defect** | 5.14 | 23.93 | |

Table : Average confusion matrix for CNN trained on experimental data.

### Simulated results:

A model was also trained on the simulated, unmodified defect response data and the same real defect free images generated from the defective test sample which were used for the experimental results. This was made up of 154 simulated defect images and 154 real defect free images sampled from the clean train dataset. After 100 training iterations, the model gave an average accuracy of 62.8%, with average F1, precision and recall scores of 0.394, 1.00 and 0.252 respectively. The average confusion matrix for the model trained on simulated data is given in Table 6.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 150 | 0 | | **No defect** | 110.74 | 37.26 | |

Table : Average confusion matrix for CNN trained on simulated data.

## Approaches for noise generation:

In this paper we explored four separate methods to map simulated data to more experimentally representative synthetic datasets by adding noise. The first approach based on using a modified CycleGAN to learn the mapping between simulated and experimental data. The second approach aims to utilize the fact that clean ultrasonic images are comparatively much more available than defect data, by combining both real clean images and defect simulations. The final two approaches studied the noise profiles seen in experimental data and attempted to simulate these at both the C scan image level and the individual A scan level.

### UT CycleGAN

To learn the mapping between simulated and experimental data, an image-to-image translation GAN was used. CycleGAN was chosen as it has shown promising results in unpaired image-to-image translation, and works particularly well for style transfer tasks which this application is similar to [29]. The fact that CycleGAN did not require paired images in training was a significant advantage as it provided greater freedom in the images used in training. Furthermore, if this approach was extended to naturally occurring defects, it would be impossible to accurately simulate the complexity of natural occurring experimental defect responses to produce a completely paired dataset.

Implementing the standard CycleGAN directly with the parameters given in the original paper[29], was unable to accurately reproduce ultrasonic images with the simulated defect responses present. Furthermore, the generated images suffered from significant mode collapse. An example of this is shown in Figure 5. Our original implementation was done in Pytorch and used: 200 epochs, a batch size of 4, 6 residual blocks, and an identity loss of 5 (half the cycle consistency loss).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CycleGAN input |  | CycleGAN output |  |
|  | Shape  Description automatically generated with low confidence |  | Shape  Description automatically generated with low confidence |  |
|  |  |  |  |  |
|  | A picture containing text, screenshot  Description automatically generated |  | A picture containing text, screenshot  Description automatically generated |  |

Figure : Example images of initial CycleGAN outputs.

#### Adjustments to CycleGAN – Mid-cycle activation map

It has been demonstrated that adjusting the loss function of CycleGAN can improve performance for specific tasks [32]. To improve the performance of the original CycleGAN paper for our task we made a variety of adjustments, the most significant being the introduction of a mid-cycle activation map loss.

The mid-cycle activation map loss aims to give the algorithm freedom to alter the noise profile whilst retaining constraint over the original defect response. The need for this is clear from the original implementation as the defect response can easily be washed out (Figure 5). To do this, we use the simulated input image to generate an activation map. This activation map is a 0-1 normalized version of the original simulated input image, this way even low amplitude defect responses are still incorporated. The simulated responses allow for this as the background response is uniform. By normalizing the activation map we zero the effect of background response and only punish the defect response. Next, a scale factor is calculated to allow for adjustments of defect size. This is calculated by taking all non-zero values (defect response) from the activation map and dividing by the total image area. We then calculate the L1 unreduced absolute error between the generated image and the simulated image. The activation map is then applied to focus the loss to the defect response and minimize the loss from the noise. This new loss map is then divided by the scale factor previously calculated from the activation map. This means that the loss function is indiscriminate of defect size and does not punish larger defects more significantly than smaller defects. Finally, the mean is taken to get the reduced value, and this is fed into the combined generator loss function given by equation 1. Figure 6 demonstrates this process with an examples image.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |
| --- |
|  |

Figure : Diagram showing how an example mid-cycle activation map loss is generated.

The mid-cycle activation map loss is only applied in the direction going from simulated responses to generated experimental images, as it relies on the clean defect response of simulated images. This is demonstrated by Figure 7.

|  |
| --- |
|  |
| (a) |
|  |
| (b) |
|  |
| (c) |

Figure : a) Our model contains two mapping functions Gexperimental : Simulated → Experimental and Gsimulated : Experimental → Simulated, and associated adversarial discriminators Dexperimental and Dsimulated. Dexperimental encourages Gexperimental to translate experimental images into outputs indistinguishable from real experimental images, and vice versa for Dsimulated and Gsimulated. Both cycles include the cycle consistency loss that was introduced in the original paper (b, c). To further encourage accurate defect reproduction, we introduce mid-cycle activation map loss for the simulated image cycle (b).

The cycle loss was also adjusted to give twice the weighting for the simulated input cycle compared to the experimental cycle. This was done to further remove restrictions on noise generation and further restrain accurate defect response. To further improve the results, t*he CycleGAN model used was adjusted from the original paper to perform better on the lower resolution 64x64 ultrasound images, by optimizing the size of the first generator convolutional layers to 3x3 instead of 7x7, with 9 residual blocks used. The GAN training data used the experimental dataset generated from the train sample, and the simulated dataset previously discussed. These were augmented using linear transformations (rotations, translations, scalar, shear, horizontal and vertical flips) to over 33000 and 12000 respective total images. The GAN model was trained over 200 epochs using a batch size of 16 using two NVIDIA GeForce RTX 3090 and took approximately 4 hours to train. The GAN model was created using the Pytorch framework.*

Once trained, the learnt mapping from the GAN was used to convert the original 154 simulated images to a new synthetic dataset of defective images. The synthetic dataset produced high quality ultrasonic images which are comparable to real experimentally obtained images, examples of the generated images are shown in Figure 5 .

|  |  |
| --- | --- |
| Generated image | Real experimental image |
|  |  |
|  |  |
| Generated image | Generated image |
|  |  |

Figure : Example of synthetic generated images and a comparative real image (top right).

#### Classification results

Training the CNN with GAN generated synthetic dataset and an equal number of clean images sampled from the clean train set, had a significant increase in classification performance when tested on the experimental clean and defective test datasets of 308 total images. After 100 training iterations, the model gave an average accuracy of 93.5%, with average F1, precision and recall scores of 0.937, 0.980 and 0.910 respectively. The average confusion matrix for the model is given in Table 7.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 143.96 | 6.03 | | **No defect** | 13.37 | 134.63 | |

Table : Average confusion matrix for CNN trained on GAN generated synthetic data.

### Real noise

Out of the 334 number of clean experimental C scan images from the clean train dataset, 154 were randomly sampled to match the size of the simulated dataset. The simulated defect images were then combined with the real noise images by summation at an individual pixel level. To not exceed the normalized upper value limit of 1, if a pixel value exceeded 1 due to the addition of noise, it was clipped to remain within the limit. This was done instead of re-normalizing the dataset as this would have reduced the noise distribution from the experimental data. From the new dataset the images where the noise was greater than the signal were removed. This left 83 final images. An example of this is demonstrated in Figure 9.

|  |  |  |
| --- | --- | --- |
| Shape, square  Description automatically generated | Shape, square  Description automatically generated | Shape, square  Description automatically generated |
| Simulated response | Real noise image | Combined synthetic image |

Figure : Example images showing the combination of real noise and simulated defect responses.

A considerable downside of the real noise approach is that it is not a fully simulated approach. This restricts its ability to scale as it requires an equal number of clean experimental images as simulated images. However, the experimental data required is from defect-free images which are considerably easier to acquire than real defect responses. The computational complexity of scaling this approach to a large number of images would be low. Therefore, if adequate clean images were available this technique could be used to produce a large dataset.

#### Classification results

Training the CNN with the real noise synthetic dataset and an equal number of clean images sampled from the clean training set had a significant increase in classification performance when tested on the experimental clean and defective test datasets. After 100 training iterations, the model gave an average accuracy of 77.4%, with average F1, precision and recall scores of 0.688, 0.950 and 0.545 respectively. The average confusion matrix for the model is given in Table 8.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 150 | 0 | | **No defect** | 67.3 | 80.7 | |

Table : Average confusion matrix for CNN trained on real noise data.

### Simulated C scan noise

To improve the on the requirement of the real noise approach needing a unique experimental image for each simulated image, we investigated if it was possible to fully simulate this noise profile. To do this, the noise distribution from the clean experimental C scan images of the defect free sample, were analyzed by plotting a histogram. It can be seen from Figure 10 that this noise profile is well aligned with an inverse gaussian distribution given by mu 0.410, loc -0.003 and scale of 0.066.

|  |
| --- |
|  |

Figure : Distribution of data from clean sample.

The simulated defect images were then combined with a noise pattern which was randomly generated for each image from an inverse gaussian distribution with the previously determined parameters. The images were combined by summation at an individual pixel level. As per the real noise method, to not exceed the normalized upper value limit of 1, if a pixel value exceeded 1 it was clipped to remain within the limit. From the new synthetic dataset the images where the noise was greater than the signal were removed, and we were left with 80 C scan final images. An example of this is demonstrated in image Figure 11.

|  |  |  |
| --- | --- | --- |
| A picture containing square  Description automatically generated | A picture containing square  Description automatically generated | A picture containing square  Description automatically generated |
| Simulated response | Generated noise image | Combined synthetic image |

Figure : Example images showing the combination of C scan simulated noise and simulated defect responses.

The implementation of C scan noise at scale would be considerably easier than the real noise approach. This is as fully simulating the noise profile from an appropriate experimental distribution requires little additional experimental data acquisition after a suitable population has been sampled. Furthermore, the computational complexity of this implementation is as efficient as the real noise approach and could scale well to produce a large dataset.

#### Classification results

Training the CNN with the C scan noise synthetic dataset and an equal number of clean images sampled from the clean training set, had a significant increase in classification performance when tested on the experimental clean and defective test datasets. After 100 training iterations, the model gave an average accuracy of 74.3%, with average F1, precision and recall scores of 0.629, 0.930 and 0.482 respectively. The average confusion matrix for the model is given in Table 5.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 150 | 0 | | **No defect** | 76.68 | 71.32 | | |

Table : Average confusion matrix for CNN trained on simulated C scan noise data.

### Simulated A scan noise

We also investigated an approach of fully generating a simulated noise profile at an A scan level which is better aligned to how noise occurs from the physical response of ultrasonic signals. For each individual time trace signal, the complete noise profile is composed of both structured noise and random noise. Structured noise are physically accurate responses, just not from a known feature. These are likely due to ply interactions and the component geometry. Random noise is independent of the samples structure and could be due to random electrical noise for example.

We assumed that for a given B scan the structural noise profile will remain constant, as for a given B scan the ply layer interactions and attenuation should be similar. Therefore, at a B scan level it is possible to remove the random noise by mean averaging the individual A scans together at each sample interval. This gives the structural noise component. For each A scan in each B scan it is then possible to work out the random noise component from the differences between each A scan and the structural noise component on a per sample basis. These combined differences can be plotted on a histogram to represent the random noise population of a B scan. This process was completed for each individual B scan. The random noise profiles were combined to give a greater number of samples for the distribution. From Figure 12 it can be seen that this distribution is well approximated by a normal distribution with 0.00 mean and a standard deviation of 0.013.

To learn the variation of the structural noise components across B scans, the average B scan structural noise was first calculated on a mean sample basis. The difference between the mean and each individual B scan structural noise profile was calculated on a per sample basis and again plotted on a histogram (Figure 13). This can be approximated by a normal distribution with mean 0.00 and standard deviation 0.003.

To generate a new noise pattern for a B scan we generate a new structural noise pattern by taking the overall mean structural noise pattern and adding variation based on the normal distribution previously calculated. To make this signal more representative of the Hilbert data, we applied a Savitzky–Golay filter to smooth the data (Figure 14). From this base signal, for each A scan a random noise profile is added following the previously determined normal distribution (Figure 12). Figure 15 helps to illustrate this process. The simulated responses were then combined with the generated combined noise profiles using a per sample summation. As per previous methods, to not exceed the normalized upper value limit of 1, if a pixel value exceeded 1 it was clipped to remain within the limit. From the new dataset the images where the noise was greater than the signal was removed, and we were left with 126 C scan final images. An example of the final images is demonstrated in Figure 16.

|  |
| --- |
| Chart, histogram  Description automatically generated |

Figure : Histogram showing the random noise distribution from the total A scans.

|  |
| --- |
|  |

Figure : Histogram showing the distribution of deviation for strucural noise from the mean structural noise pattern.

|  |
| --- |
|  |
| Chart, line chart  Description automatically generated |

Figure : An example of how a structural noise profile is generated from the mean.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Example B scan of strucural noise | Example B scan of random noise | Example B scan of combined noise profiles |

Figure : An example of how structural and random noise profiles are combined at a B scan level.

|  |  |  |
| --- | --- | --- |
| Chart  Description automatically generated | Shape  Description automatically generated | Chart, histogram  Description automatically generated |
| Simulated response | Generated noise image | Combined synthetic image |

Figure : Example images showing the combination of A scan simulated noise and simulated defect responses.

Implementing the A scan noise profile is a fully simulated approach. However, it requires a greater level of analysis compared to the C scan level noise method before implementation. Furthermore, as the generation of the noise pattern is required on a per B scan level, an additional computational step is required to cover the number of B scans. This is therefore less computationally efficient than both the real noise and C scan noise implementation.

#### Classification results

Training the CNN with the A scan noise synthetic dataset and an equal number of clean images sampled from the clean training set, had a significant increase in classification performance when tested on the experimental test datasets. After 100 training iterations, the model gave an average accuracy of 80.0%, with average F1, precision and recall scores of 0.738, 0.970 and 0.598 respectively. The average confusion matrix for the model is given in Table 6.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **True \ Predicted** | **Defect** | **No defect** | | **Defect** | 150 | 0 | | **No defect** | 59.48 | 88.52 | |

Table : Average confusion matrix for CNN trained on simulated A scan noise data.

# Discussion and comparison

## Comparison of results

Figure 17 shows examples of images produced by the different synthetic data generation methods. The classification results are summarized in Figure 18, which shows the mean accuracy and F1 score for each dataset investigated.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Shape, square  Description automatically generated | A picture containing square  Description automatically generated | Chart, histogram  Description automatically generated |
| **GAN generated** | **Real noise** | **C scan simulated noise** | **A scan simulated noise** |

Figure : Comparison of different synthetic data image examples.

|  |
| --- |
|  |

Figure : Comparison of classification results for each dataset.

## Model interpretability with Grad-CAM

A key barrier to the uptake of Machine Learning in NDT is a lack of model interpretability [18]. The use of synthetic data has the potential to further mystify this process. To help combat this we implemented Guided Gradient-weighted Class Activation Mapping (Guided Grad-CAM) for a randomly selected model trained from each dataset, evaluated on experimental data. Guided Grad-CAM is a technique for producing ‘visual explanations’ of CNNs with the goal of making them more transparent and explainable [33]. Guided Grad-CAM gives an indication of how the model interprets the data and has been shown to help users place greater trust in a model. The goal here is to help visualize if the models trained on synthetic data are using similar features for prediction compared to experimental data. Figure 16 shows the defective experimental test image, and both the associated Guided Grad-CAM image, and a mixed image which combines the Guided Grad-CAM and the input image with a respective weighting of 1.5.

|  |
| --- |
|  |
| **Model trained on experimental data** |
|  |
| **Model trained on real noise synthetic data** |
|  |
| **Model trained on C scan noise synthetic data** |
|  |
| **Model trained on A scan noise synthetic data** |

Figure : Example of Grad-CAM visualization of models trained on different datasets.

## Discussion

Simulated UT data of defect responses in composites lacks the complexity of experimental noise. In this work we have demonstrated that this has a significant negative impact on classification performance when CNN classifiers are trained on simulated data and tested on real data, with an average F1 score of 0.39. However, we have demonstrated that it is possible to reduce this effect by creating synthetic datasets which aim to better simulate real experimental data. We have explored four different methods to create synthetic datasets, but they have all shown significant increases in classification performance compared to the original simulated dataset. The modified CycleGAN generated synthetic dataset produced significantly better classification results than the other methods, with an Average F1 score of 0.94. This exceeded the classifier trained on a subset of the experimental dataset, but this may is likely due to the reduction in available training data due to the test split and should not be considered a direct comparison. Real noise, simulated C scan noise, and simulated A scan noise produced similar mean accuracy results, but the simulated A scan noise synthetic dataset produced the best average F1 score of the three, with 0.74. It is interesting that the simulated A scan noise dataset outperformed the real noise synthetic dataset. This may be due to the fact the real noise obscures the defect response features too much.

These results demonstrate that in scenarios where noisy experimental environments can cause real data to vary greatly from simulated data, synthetic datasets provide an opportunity for more effective training data. This is particularly beneficial as we retain the accuracy and fully labelled nature of physics-based simulations, which allow us to fully control the simulation of different defect class types and the variability within them.

Alongside the ability to accurately simulate noise response, a further reason for improved classification results for GAN and A scan synthetic datasets may be their ability to account for depth wise signal attenuation and adjust the noise levels with respect to depth. This produces more appropriate noise levels for deeper and weaker defect responses and allows for the preservation of many more simulated responses. Unlike simulated C scan and real noise approaches which are defect depth agnostic and therefore result in the rejection of more images due to the concealment of low-level responses with consistent noise at all response depths.

When considering the broader aim of generating large synthetic datasets that could be used to create a database of realistic training examples, it is important to consider the ease and robustness of synthetic data generation. Training of the CycleGAN is a delicate process and whilst it has been able to produce realistic images for Flat-Bottom Holes, it may struggle to generalize to other defects without significantly broader examples of defects in training. This would largely defeat the point of the synthetic data generation in this instance. Furthermore, the training of an effective GAN model is still extremely challenging and the process of hyperparameter selection is not robust. It is therefore favorable to consider an approach that is robust to different defects and can be scaled. For scalability, a fully simulated method is preferable to a method which still requires significant collection of experimental data. Therefore, the real noise approach is superseded by both the A scan and C scan synthetic approaches. The C scan noise approach is slightly easier to implement than the A scan as it requires less experimental data analysis and can be done at the C scan image level instead of the A scan level. Further work could be done to explore the distribution of C scan noise at different depths to enable maintenance of a larger number of simulated responses in a simpler way than the complex A scan noise simulation method. This could potentially combine the benefits of both the A scan and C scan noise approaches.

It has been identified in literature that model interpretability is a key limiting factor in the uptake of DL in NDT. Guided Grad-CAM was implemented to try and minimize the obscurity that using synthetic data could produce. Whilst model interpretability is a complex field of research and interpretability is challenging to quantify, we believe that the Guided Grad-CAM results at least indicate that models trained on synthetic data are learning similar features compared to models trained on purely experimental data. This is demonstrated since each Grad-CAM image correctly highlights defect pixels only for defect detection. This is very encouraging as it helps to give confidence over the use of synthetic data when training DL models for NDT.

# Conclusion

In this work we addressed the problem of using simulated ultrasonic NDT data where the simulation results, whilst physically accurate, can vary substantially from real-world data due to experimental complications such as random and structural noise. With a modified loss function to encourage accurate defect response, CycleGAN proved a suitable candidate for this task, allowing us to maintain the utility of simulating data from physics-based models and convert them to more experimentally realistic synthetic datasets. The CycleGAN synthetically produced dataset showed the greatest improvement in classification performance. Along with CycleGAN, other methods for introducing simulated and real experimental noise were investigated. These methods also showed an improvement to classification performance, in addition to being easier to implement and more robust than CycleGAN, even with the modifications.

Whilst the classification results were not perfect, this work demonstrates that the synthetic data generation methods were able to successfully transfer the simulation domain closer to the experimental domain. In future work, we hope to investigate if HPO on a synthetic dataset can improve its real data classification performance to produce a more accurate classifier. Additionally, the next steps in this work will look to see if the style transfer can be extended across the full range of defect types. If successful, large, fully annotated, synthetic datasets could be efficiently produced, opening the potential for further use of Deep Learning in NDT.

# Acknowledgment

This work was supported through Spirit AeroSystems/ Royal Academy of Engineering Research Chair for In-Process Non-Destructive Testing of Composites, RCSRF 1920/10/32

# References

[1] B. Djordjevic, ‘Non Destructive Test Technology for the Composite’, p. 7, Jan. 2009.

[2] C. Meola, S. Boccardi, G. M. Carlomagno, N. D. Boffa, E. Monaco, and F. Ricci, ‘Nondestructive evaluation of carbon fibre reinforced composites with infrared thermography and ultrasonics’, *Compos. Struct.*, vol. 134, pp. 845–853, Dec. 2015, doi: 10.1016/j.compstruct.2015.08.119.

[3] A. M.-E. Dorado, ‘Composite Material Characterization using Acoustic Wave Speed Measurements’, p. 5.

[4] Ley, O. and V. Godinez, ‘Non-destructive evaluation (NDE) of aerospace composites: application of infrared (IR) thermography’, doi: 10.1533/9780857093554.3.309.

[5] A. Kokurov and D. Subbotin, ‘Ultrasonic detection of manufacturing defects in multilayer composite structures’, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1023, p. 012013, Jan. 2021, doi: 10.1088/1757-899X/1023/1/012013.

[6] D. K. Hsu, ‘15 - Non-destructive evaluation (NDE) of aerospace composites: ultrasonic techniques’, in *Non-Destructive Evaluation (NDE) of Polymer Matrix Composites*, V. M. Karbhari, Ed. Woodhead Publishing, 2013, pp. 397–422. doi: 10.1533/9780857093554.3.397.

[7] F. Heinecke and C. Willberg, ‘Manufacturing-Induced Imperfections in Composite Parts Manufactured via Automated Fiber Placement’, *J. Compos. Sci.*, vol. 3, no. 2, p. 56, Jun. 2019, doi: 10.3390/jcs3020056.

[8] I. Papa, V. Lopresto, and A. Langella, ‘Ultrasonic inspection of composites materials: Application to detect impact damage’, *Int. J. Lightweight Mater. Manuf.*, vol. 4, no. 1, pp. 37–42, Mar. 2021, doi: 10.1016/j.ijlmm.2020.04.002.

[9] L. Séguin-Charbonneau, J. Walter, L.-D. Théroux, L. Scheed, A. Beausoleil, and B. Masson, ‘Automated defect detection for ultrasonic inspection of CFRP aircraft components’, *NDT E Int.*, vol. 122, p. 102478, Sep. 2021, doi: 10.1016/j.ndteint.2021.102478.

[10] S. Gholizadeh, ‘A review of non-destructive testing methods of composite materials’, *Procedia Struct. Integr.*, vol. 1, pp. 50–57, 2016, doi: 10.1016/j.prostr.2016.02.008.

[11] M. Jolly *et al.*, ‘Review of Non-destructive Testing (NDT) Techniques and their Applicability to Thick Walled Composites’, *Procedia CIRP*, vol. 38, pp. 129–136, Jan. 2015, doi: 10.1016/j.procir.2015.07.043.

[12] S. Maack, V. Salvador, and S. David, ‘Validation of artificial defects for Non-destructive testing measurements on a reference structure’, *MATEC Web Conf.*, vol. 199, p. 06006, Jan. 2018, doi: 10.1051/matecconf/201819906006.

[13] P. Gardner *et al.*, ‘Machine learning at the interface of structural health monitoring and non-destructive evaluation’, *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 378, no. 2182, p. 20190581, Oct. 2020, doi: 10.1098/rsta.2019.0581.

[14] ‘Introduction to non-destructive testing’, *Aerospace Testing International*, Oct. 25, 2018. https://www.aerospacetestinginternational.com/features/introduction-to-non-destructive-testing.html (accessed Nov. 17, 2021).

[15] F. W. Margrave, K. Rigas, D. A. Bradley, and P. Barrowcliffe, ‘The use of neural networks in ultrasonic flaw detection’, *Measurement*, vol. 25, no. 2, pp. 143–154, Mar. 1999, doi: 10.1016/S0263-2241(98)00075-X.

[16] J. Ye, S. Ito, and N. Toyama, ‘Computerized Ultrasonic Imaging Inspection: From Shallow to Deep Learning’, *Sensors*, vol. 18, no. 11, p. 3820, Nov. 2018, doi: 10.3390/s18113820.

[17] B. Valeske, A. Osman, F. Römer, and R. Tschuncky, ‘Next Generation NDE Sensor Systems as IIoT Elements of Industry 4.0’, *Res. Nondestruct. Eval.*, vol. 31, no. 5–6, pp. 340–369, Nov. 2020, doi: 10.1080/09349847.2020.1841862.

[18] S. Cantero-Chinchilla, P. D. Wilcox, and A. J. Croxford, ‘Deep learning in automated ultrasonic NDE -- developments, axioms and opportunities’, *ArXiv211206650 Eess*, Dec. 2021, Accessed: Jan. 12, 2022. [Online]. Available: http://arxiv.org/abs/2112.06650

[19] M. Meng, Y. J. Chua, E. Wouterson, and C. P. K. Ong, ‘Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks’, *Neurocomputing*, vol. 257, pp. 128–135, Sep. 2017, doi: 10.1016/j.neucom.2016.11.066.

[20] B. Wang, Y. Li, Y. Luo, X. Li, and T. Freiheit, ‘Early Event Detection in a Deep-learning Driven Quality Prediction Model for Ultrasonic Welding’, *J. Manuf. Syst.*, vol. 2021, pp. 325–336, Jun. 2021, doi: 10.1016/j.jmsy.2021.06.009.

[21] R. J. Pyle, R. L. T. Bevan, R. R. Hughes, R. K. Rachev, A. A. S. Ali, and P. D. Wilcox, ‘Deep Learning for Ultrasonic Crack Characterization in NDE’, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 68, no. 5, pp. 1854–1865, May 2021, doi: 10.1109/TUFFC.2020.3045847.

[22] S. Lonne, S. Mahaut, and G. Cattiaux, ‘EXPERIMENTAL VALIDATION OF CIVA ULTRASONIC SIMULATIONS’, Jan. 2006.

[23] M. Darmon *et al.*, ‘VALIDATION OF AN ULTRASONIC CHARACTERIZATION TECHNIQUE FOR ANISOTROPIC MATERIALS: COMPARISON OF EXPERIMENTS WITH BEAM PROPAGATION MODELLING’, p. 20.

[24] K. Jezzine, D. Ségur, R. Ecault, and N. Dominguez, ‘Simulation of ultrasonic inspections of composite structures in the CIVA software platform’, p. 8.

[25] A. Figueira and B. Vaz, ‘Survey on Synthetic Data Generation, Evaluation Methods and GANs’, *Mathematics*, vol. 10, no. 15, Art. no. 15, Jan. 2022, doi: 10.3390/math10152733.

[26] A. Antoniou, A. Storkey, and H. Edwards, ‘Data Augmentation Generative Adversarial Networks’, *ArXiv171104340 Cs Stat*, Mar. 2018, Accessed: Feb. 03, 2022. [Online]. Available: http://arxiv.org/abs/1711.04340

[27] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, ‘GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification’, *Neurocomputing*, vol. 321, pp. 321–331, Dec. 2018, doi: 10.1016/j.neucom.2018.09.013.

[28] S. Motamed, P. Rogalla, and F. Khalvati, ‘Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images’, *Inform. Med. Unlocked*, vol. 27, p. 100779, Jan. 2021, doi: 10.1016/j.imu.2021.100779.

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, ‘Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks’. arXiv, Aug. 24, 2020. Accessed: May 24, 2022. [Online]. Available: http://arxiv.org/abs/1703.10593

[30] D. Medak, L. Posilovic, M. Subasic, M. Budimir, and S. Loncaric, ‘Automated Defect Detection From Ultrasonic Images Using Deep Learning’, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 68, no. 10, pp. 3126–3134, Oct. 2021, doi: 10.1109/TUFFC.2021.3081750.

[31] C. Mineo *et al.*, ‘Flexible integration of robotics, ultrasonics and metrology for the inspection of aerospace components’, *AIP Conf. Proc.*, vol. 1806, no. 1, p. 020026, Feb. 2017, doi: 10.1063/1.4974567.

[32] C. D. Walsh, J. Edwards, and R. H. Insall, ‘Ensuring accurate stain reproduction in deep generative networks for virtual immunohistochemistry’. arXiv, Apr. 14, 2022. Accessed: Aug. 25, 2022. [Online]. Available: http://arxiv.org/abs/2204.06849

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ‘Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization’, *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.