

# **IRT-GAN: A generative adversarial network with a multi-headed fusion strategy for automated defect detection in composites using infrared thermography**

Liangliang Cheng<sup>1\*</sup>, Zongfei Tong<sup>1,2</sup>, Shejuan Xie<sup>2</sup> and Mathias Kersemans<sup>1\*</sup>

<sup>1</sup>Mechanics of Materials and Structures (MMS), Ghent University, Technologiepark 46, B-9052 Zwijnaarde, Belgium.

<sup>2</sup>State Key Laboratory for Strength and Vibration of Mechanical Structures, Xi'an Jiaotong University, 710049, Xi'an, China.

\*Corresponding authors: liangliang.cheng@ugent.be; mathias.kersemans@ugent.be

**Abstract:** InfraRed Thermography (IRT) is a valuable diagnostic tool to non-destructively detect defects in fiber reinforced polymers. Often, a range of processing techniques are applied, e.g. principal component analysis, Fourier transformation, and thermographic signal reconstruction, in an attempt to enhance the defect detectability. Still, for the actual defect detection and evaluation, the interpretation by an expert operator is required which thus limits the (industrial) application potential of infrared thermography.

This study proposes a Generative Adversarial Network (GAN) framework, termed IRT-GAN, to create a single unique thermal-image-to-segmentation translation of defects in composite materials. A large augmented numerical dataset has been simulated for a range of composite materials with different defects in order to train the IRT-GAN model. Integrated with the Spatial Group-wise Enhance layer, the IRT-GAN takes six pre-processed thermal images, thermographic signal reconstruction images in our case, as input and progressively fuses them via a multi-headed fusion strategy in the Generator. As such, this proposed IRT-GAN framework leads to the automated generation of a unique defect segmentation image.

The high performance of the IRT-GAN, trained on the virtual dataset, is demonstrated on experimental data of both glass and carbon fiber reinforced polymers with various defect types, sizes, and depths. In addition, it is investigated how early, middle, and late-stage feature fusion in the GAN influences the segmentation performance.

**Keywords:** Deep learning, GAN, Infrared thermography, Composite, Defect detection, Non-destructive testing, Image fusion

## 1. Introduction

As an important structural material today, Carbon Fiber Reinforced Polymer (CFRP) and Glass Fiber Reinforced Polymer (GFRP) composites are used in a wide range of applications, with particular relevance to the aerospace and automotive industries. Inevitably, composite materials suffer from defects, such as delamination and porosity, due to the production process and/or unacceptable operating loads. Therefore, Non Destructive Testing (NDT) has emerged as a critical technology to guarantee the structural integrity of composite components.

Active InfraRed Thermography (IRT) is an appealing NDT technique for diagnostic purposes that utilizes an InfraRed (IR) camera to rapidly and accurately measure the thermal response for the purpose of detecting and quantifying defects [1-3]. However, the recorded raw data from the IR camera is frequently contaminated by a variety of noise sources, such as external reflections, variations in the optical properties of the specimen, non-uniform heating, and instrumental noise. And the consequence in this regard is a certain degree of impairment in defect detection. To mitigate the various noise effects and to enhance the detectability of defects in composite materials, several post-processing techniques have been developed in recent years, such as Thermographic Signal Reconstruction (TSR) [4-6], Principal Component Thermography (PCT) [7-8], Independent Component Thermography (ICT) [9-10] and Pulsed Phase Thermography (PPT) [11-12]. Through a low-order polynomial function, TSR fits the surface temperature evolution at each pixel during the cooling-down phase. Typical polynomial orders of four to seven are used [13–15], yielding five to eight images representing the corresponding polynomial coefficients. Typically, one TSR coefficient image fails to capture all defects, and the final solution for identifying defects is usually based on a comprehensive evaluation of all TSR images. The PCT technique projects the 3D thermal response data into an orthogonal space using Principal Component Analysis. The resulting Principal Component images can explain the largest part of the variability in the 3D thermal response data. However, it is not straightforward to predict which PC image captures the variance due to the presence of defects because this depends on the size, number, depth of defects, and the background noise level. In PPT, the harmonic components of the thermal response evolution are extracted in a pixel-wise manner using Fourier decomposition, and the phase contrast is evaluated. Depending on the defect type and defect depth, different evaluation frequencies provide optimal defect detectability.

What all these techniques have in common is that they compress the recorded IR dataset into several representative images, aiming at reducing background noise and improving defect

detectability. But even then, the selection and interpretation of the post-processed images, in order to detect the presence of defects, is not always straightforward. Often, this still requires human intervention and experienced experts in order to properly detect and evaluate defect features. It would be more interesting if the human factor in selecting and interpreting post-processed thermal images could be excluded, as such to further enhance the defect detectability. The present study proposes a deep learning image segmentation model, termed IRT-GAN, for automated analysis of thermography data. The IRT-GAN involves an image fusion strategy under the framework of Generative Adversarial Networks (GANs) [16-18], and results in a single segmented image revealing the presence of defects. Although image segmentation techniques have been widely used in natural image processing, rare applications in infrared thermography non-destructive testing have been reported so far [19-22]. Different from the deep learning approaches in [19-22], our approach focuses on the post-processed images such as TSR, PCT or PPT images, and fuses them into a unique segmentation image via the proposed IRT-GAN model. Considering that the origin of the thermographic data is of minor importance for our deep learning framework, it can be applied to a range of test modalities employing different heat excitation waveforms, e.g. flash heating [23], step heating [24], square wave heating [25] and frequency and/or phase modulated heating [26].

The novelties of the proposed IRT-GAN segmentation model are highlighted :

- 1) A customized generator and discriminator architecture are designed to accomplish the goal of detecting defects.
- 2) The training of the proposed IRT-GAN model is solely conducted on the basis of a virtual dataset (no involvement of experimental data), and it is afterwards tested on experimental data.
- 3) It generates a single unique defect segmentation image with improved defect detectability, thereby avoiding any human intervention and/or threshold selection.

The paper is organized as follows: Section 2 introduces the proposed IRT-GAN framework. Section 3 describes the preparation of training and test datasets. Section 4 illustrates the image segmentation results on CFRP and GFRP panels with a range of defects and further discusses two alternative fusion strategies. Section 5 concludes the study and outlines the future work.

## **2. Methodology**

### **2.1 The introduction of GAN**

Ian Goodfellow's article on the GAN [16] framework, first published in 2014, has drawn substantial attention from researchers working in Machine Learning (ML), as evidenced by the

constant emergence of new frameworks and applications based on GAN [27-33]. This technique can be thought of as a novel synthesis of game and graph theory, which essentially consists of two adversarial models: Generator  $G$  and Discriminator  $D$ . The principal function of the generator  $G$  is to capture the distribution of real images to deterministically generate new fake but plausible images, thus deceiving the discriminator  $D$ . The discriminator  $D$  need to ascertain the genuineness of the received images (real or fake). The objective of the training process is to achieve Nash equilibrium [16], which indicates that  $G$  has successfully learned the distribution of real data and generated plausible fake data, while  $D$  has lost the ability to distinguish anymore between real and fake data. The schematic diagram of GAN is shown in Fig.1.

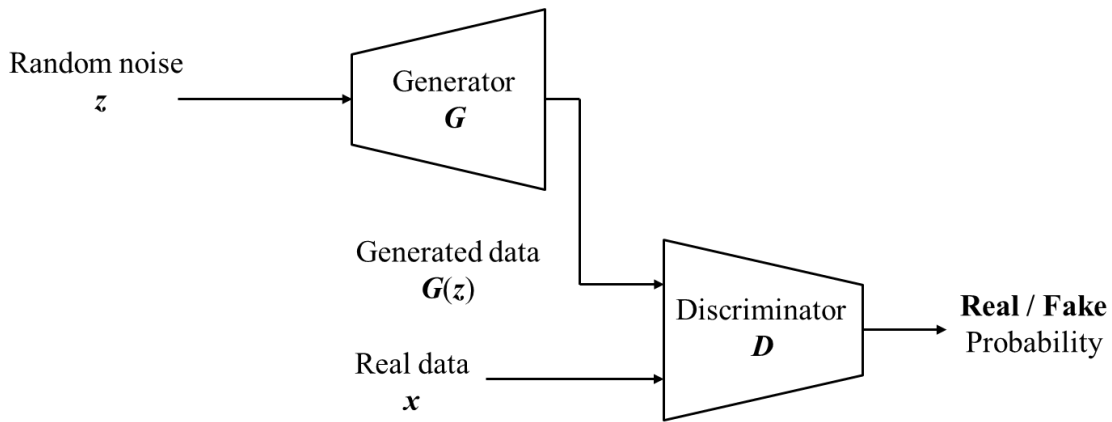


Fig.1 The schematic diagram of GAN

The objective function of GAN is expressed as below:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

In which  $E$  represents the expectation value,  $x$  is the real data sampled from  $p_{data}$ , and  $z$  is input noise variable sampled from a prior defined distribution  $p_z$ , for instance, a one-dimensional Gaussian distribution. The GAN's training is achieved via the optimization of those two terms in equation (1), representing the entropy of the data from the real distribution and the learned distribution from  $G$  judged by  $D$ , respectively. More specifically, the goal of training  $D$  is to maximize the objective function (1) by assigning the probability scores of real and generated data. The training target of  $G$  is entirely opposite as it wants to minimize the objective function (1), which is equivalent to increasing the probability score of generated data judged by  $D$ . For more details on GAN theory, readers are referred to the original GAN paper [16].

Owing to the several distinct advantages of GAN, such as no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples,

several successful applications in various fields have emerged recently, including image-to-image translation [27], super-resolution images [28-29], image inpainting [30-31], and classification [32-33].

## 2.2 Proposed IRT-GAN framework

One of the primary drawbacks of IRT defect detection approaches based on the post-processing techniques such as PCT, TSR, and PPT lies in the diversity and uncertainty of generated outputs, which necessitates human intervention in the interpretation with respect to defect detection and also complicates fusion techniques in deep learning. In order to overcome this, we propose a deep learning framework IRT-GAN which can be applied on post-processed images (e.g. TSR, PCT, PPT) with the goal of obtaining a single unique segmentation result and an enhanced defect detectability.

The original pix2pixGAN model consists of a generator (U-Net [34]) and a discriminator (Patch GAN) and is applied to the general purpose of end-to-end translation [27]. However, the single-input-single-output mode confines our case due to the fact that defects in composites are likely to be present in more than just a single post-processed TSR/PCT/PPT image. Inspired by pix2pixGAN, we propose an IRT-GAN architecture founded on the pix2pixGAN model and designed for the defect segmentation task in composite materials. The architecture of IRT-GAN still retains the fundamental skeleton of pix2pixGAN but adapts the generator and discriminator to enhance defect detectability. Specifically, a multi-headed generator is implemented in order to fuse the post-processed thermal images, i.e., the TSR, PCT, or PPT images, while a two-path discriminator (GlobalGAN and PatchGAN) [30] is employed in order to not only capture the holistic features but also the local continuity in the images.

The following section demonstrates the formulation and framework of IRT-GAN, followed by a discussion of the structures of  $\mathbf{G}$  and  $\mathbf{D}$ . By the end of this section, the objective function is analysed.

### 2.2.1 The overall structure

This part demonstrates IRT-GAN at the general architecture level, as displayed in Fig.2. The IRT-GAN contains a multi-headed generator and two-path discriminator. The original U-Net contains a contraction path stacked with multiple convolutions and max-pooling layers used to downsample for generating high-level features (bottleneck) and an expanding path used to enable precise localization using transposed convolution. In particular, skip-connection as a crucial operation in U-Net is used to concatenate the outputs from the decoder to the feature maps from the encoder at each step, thus recovering better fine-grained details in the prediction.

It has already demonstrated its superiority in segmentation tasks for medical images [35-36]. As a result, the generator in IRT-GAN employs U-Net as the basic structure, consisting of an encoder with multiple encoding paths and a decoder. To summarize, the primary training steps are as follows: First, numerical IRT datasets are generated using a simulation model (see section 3.1), providing a way to obtain a large virtual training database with ground-truth defect labels. Then, TSR (can also be replaced by PCT or PPT) analysis is conducted on the simulated IRT datasets. For the TSR technique, a 5<sup>th</sup> order polynomial is adopted to fit pixel-by-pixel the temporal evolution of the numerically simulated IRT datasets. This results in six TSR coefficient images denoted as Image 1 - Image 6 in Fig.2, for each simulation case. Next, the TSR images are fed into the multi-headed generator to get fused feature maps in the encoder and to generate a unique segmentation image from the decoder together with the strategy of skip connections from the encoder. Then, the segmentation image and ground-truth image are concatenated with the raw 6 TSR images, forming a 21-channel image matrix (each TSR contains 3 RGB channels, plus 3 channels from the ground truth image, thus resulting in a total 21-channel matrix after concatenating them) and fed into the discriminator to make a judgment on their authenticity. Last, the parameters in the generator and discriminator are updated via the backpropagation mechanism according to a predefined loss function.

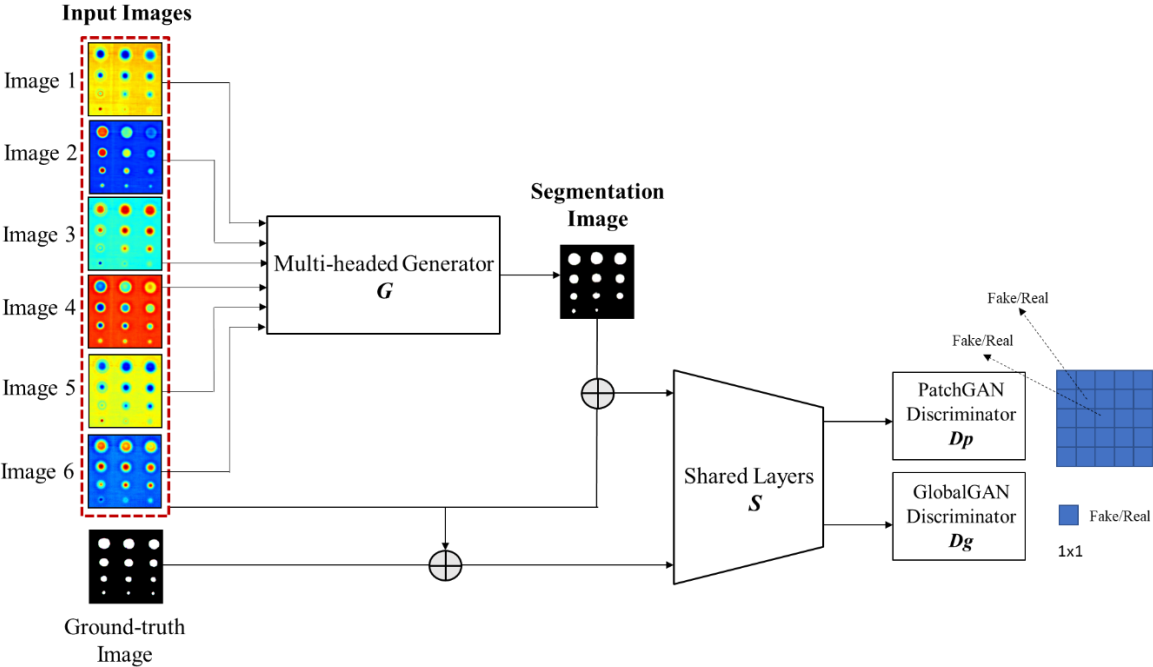


Fig.2 The overall architecture of IRT-GAN

### 2.2.2 Generator network

One of the challenges of multi-modal image fusion in deep learning is to optimally extract and fuse individual core information from each single (monomodal) image. As the core of the proposed IRT-GAN, taking the U-Net architecture as the backbone with an encoder and decoder, the multi-headed generator  $G$  is composed of convolutional blocks ( $C$ ), Transposed Convolutional blocks ( $TCN$ ), and Spatial Group-wise Enhance blocks ( $SGE$ ) [37]. Fig.3 shows an in-depth view of the architecture of the generator. Among them, the  $C$  block is formed by linking a convolutional layer [38], a BatchNorm layer [39], a LeakyReLU layer [40], and a dropout layer [41], which are used to downsample and to extract feature maps.  $TC$  block is designed to be similar to the  $C$  block, in which the convolutional layer and the LeakyReLU layer are replaced as a transposed-convolutional layer [42] and a ReLU layer [43], respectively. Putting these blocks in the appropriate order with skip-connections can form a network to realize an adversarial training process. Specifically, the architecture of the training flow for the generator is presented in Fig.3.

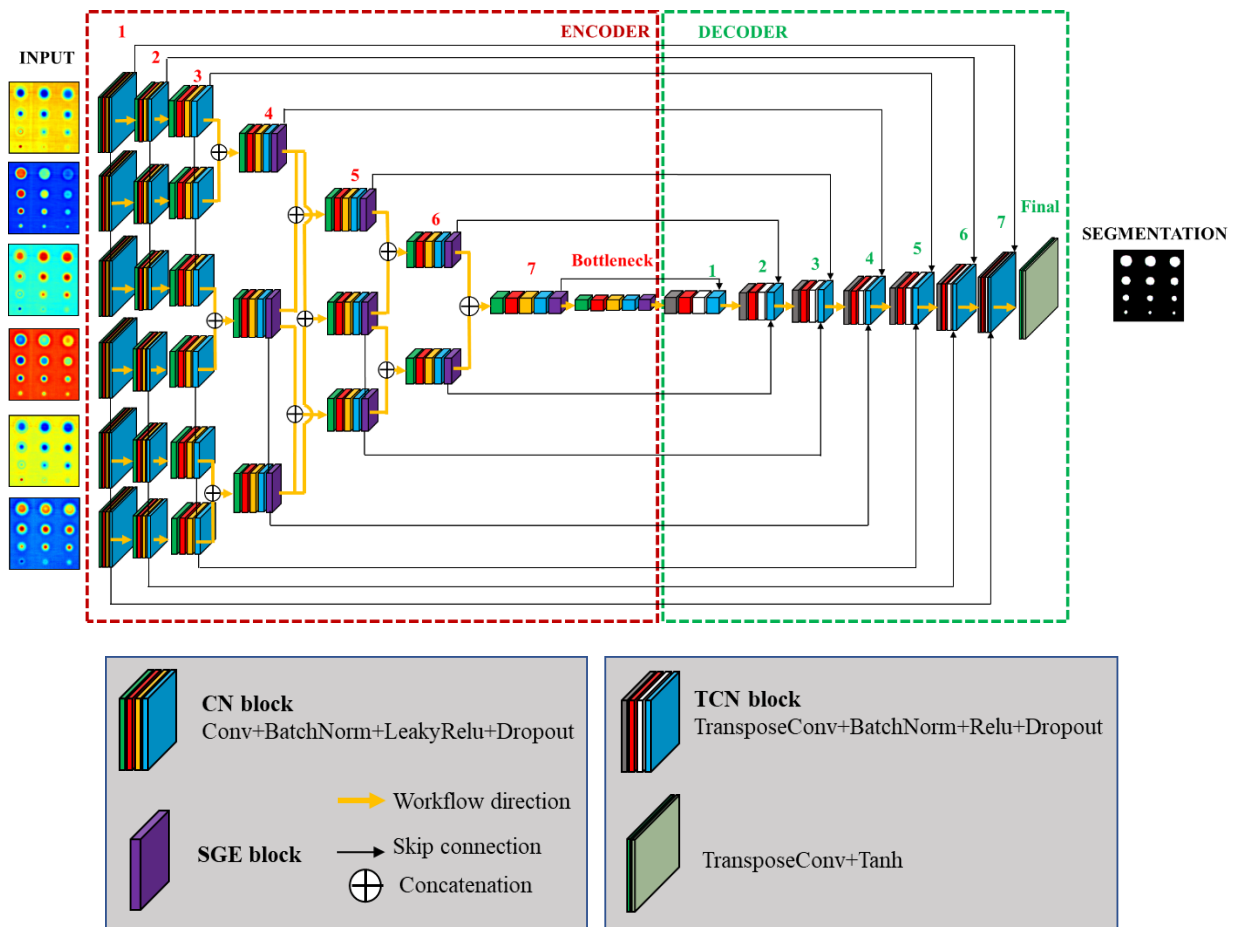


Fig.3 The architecture of the generator  $G$

- **Encoder**

An encoder is a network that receives an input and delivers a condensed feature vector or map. These vectors holding representative information can ideally be mapped to the original input. The input of the generator is, in our case, six TSR images with an image size of  $256 \times 256 \times 3$  (Width  $\times$  Height  $\times$  Channel). Before demonstrating the technical details, a couple of notions are first introduced.

### *Convolutional block (C)*

Convolutional neural networks are able to produce image representations that capture hierarchical patterns and attain global theoretical receptive fields. And one of the popular convolutional blocks  $C$  in deep learning frameworks consists of convolutional, BatchNorm, leakyReLU, and Dropout layers consecutively in sequence as shown in Fig.4, in which the BatchNorm layer serves to stabilize the learning process by setting the units to have zero mean and unit variance, while leakyReLU allows for a small, non-zero gradient when the unit is not active, and the dropout layer is designed to avoid over-fitting. In addition, the  $C$  module also serves to transform the original feature map  $X$  into a new feature map  $X_1$  with dimensionality reduction from  $W1 \times H1 \times C1$  to  $W2 \times H2 \times C2$ , as indicated in Fig.4.

### **C module**

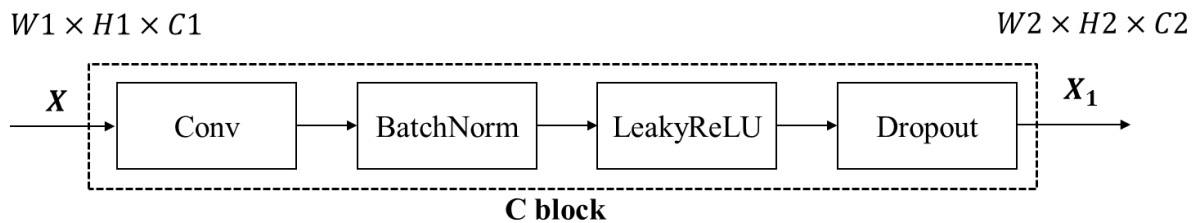


Fig.4 The schema of the  $C$  module

### *Spatial Group-wise Enhance (SGE)*

In order to suppress noise and further enhance the learned feature maps, a Spatial Group-wise Enhance Network ( $SGE$ ) [37] is being employed on top of a  $C$  block, adaptively adjusting the weights of each sub-feature by generating an attention factor for each spatial location in each semantic group. It is worth mentioning that the  $SGE$  module is very lightweight and, by its nature, requires almost no additional parameters or calculations. Its schema is illustrated in Fig.5.



### C-SGE module

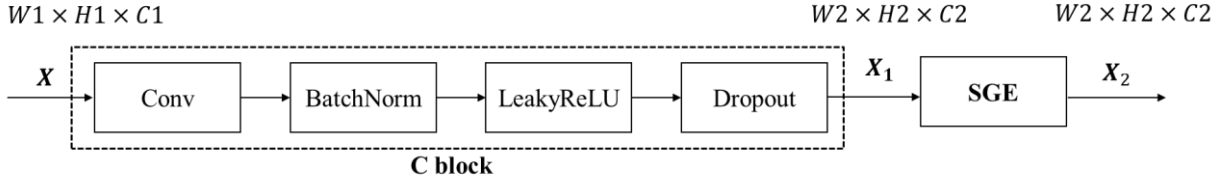


Fig.5 The schema of the *C-SGE* module

From Fig.5, it can be seen the dimensionality of the output  $X_2$  from the *SGE* block remains the same as  $X_1$ . For more details about the *SGE* block, readers are referred to [37].

### Encoder architecture

The encoder in the generator first takes six TSR images as input and passes them into individual *C* blocks with convolutional kernels  $4 \times 4$ . All strides and paddings are set to 2 and 1, respectively. The padding mode is set as “reflect” to avoid causing undesirable grey edges. The first *C* block is designed to extract the feature maps from each input image independently and keeps the outputs from the convolution blocks at half the size of the inputs. This operation can be expressed using the following equation.

$$EO_k^l = C(EO_k^{l-1}, EW_k^l) \quad (2)$$

Equation (2) depicts the nonlinear mapping from original input into feature maps, where  $EO_k^l$ ,  $EW_k^l$  denote the output feature map and the weights of the  $l^{\text{th}}$  block layer from the  $k^{\text{th}}$  encoding path in the encoder.  $EO^0$  stands for the original input images. The numbers in Fig.3 indicate the number of the various block layers.

The feature maps of each input image are learned and extracted independently via the first three layers of the encoder, before being further processed for the fusion purpose. The implementation of this process can be described via equations (3)-(5).

$$EO_k^1 = C(EO_k^0, EW_k^1) \quad (3)$$

$$EO_k^2 = C(EO_k^1, EW_k^2) \quad (4)$$

$$EO_k^3 = C(EO_k^2, EW_k^3) \quad k = 1,2,3,4,5,6 \quad (5)$$

The learned feature maps  $EO^3$  are then fed into the fusion procedure. One of the standard fusion operations of feature maps is to concatenate the feature maps at bottleneck level or after several convolution layers, yet this involves a significant number of additional parameters to use in

training. To alleviate and successfully handle the complexity of the training process, the feature fusion blocks in the encoder are sparsely connected. More specifically, channel-wise concatenation is first applied to  $\mathbf{EO}_1^3$  and  $\mathbf{EO}_2^3$ , with a  $\mathbf{C}$  block appended to obtain a fused feature map  $\mathbf{EO}_1^4$ . Likewise, the fused feature maps  $\mathbf{EO}_2^4$  and  $\mathbf{EO}_3^4$  can be obtained in the same manner. This operation can be described by equations (6)-(8).

$$\mathbf{EO}_1^4 = \mathbf{C}(\text{concat}(\mathbf{EO}_1^3, \mathbf{EO}_2^3), \mathbf{EW}_1^4) \quad (6)$$

$$\mathbf{EO}_2^4 = \mathbf{C}(\text{concat}(\mathbf{EO}_3^3, \mathbf{EO}_4^3), \mathbf{EW}_2^4) \quad (7)$$

$$\mathbf{EO}_3^4 = \mathbf{C}(\text{concat}(\mathbf{EO}_5^3, \mathbf{EO}_6^3), \mathbf{EW}_3^4) \quad (8)$$

The extracted fused feature maps  $\mathbf{EO}_1^4$ ,  $\mathbf{EO}_2^4$  and  $\mathbf{EO}_3^4$  are continually fed into the next fusion layer composed of a  $\mathbf{C}$ - $\mathbf{SGE}$  module. And this procedure can be formulated as

$$\mathbf{EO}_1^5 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_1^4, \mathbf{EO}_2^4), \mathbf{EW}_1^5), \mathbf{EW}_1^{5-SE}) \quad (9)$$

$$\mathbf{EO}_2^5 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_1^4, \mathbf{EO}_3^4), \mathbf{EW}_2^5), \mathbf{EW}_2^{5-SE}) \quad (10)$$

$$\mathbf{EO}_3^5 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_2^4, \mathbf{EO}_3^4), \mathbf{EW}_3^5), \mathbf{EW}_3^{5-SE}) \quad (11)$$

In which  $\mathbf{W}_k^{l-SE}$  means the weights of the  $l^{\text{th}}$  block layer from the  $k^{\text{th}}$  encoding path involved in the  $\mathbf{SGE}$  layer. The  $\mathbf{SGE}$  block appended to the  $\mathbf{C}$  block enables the quality improvement of the feature representations via a learnable attention mechanism.

$$\mathbf{EO}_1^6 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_1^5, \mathbf{EO}_2^5), \mathbf{EW}_1^6), \mathbf{EW}_1^{6-SE}) \quad (12)$$

$$\mathbf{EO}_2^6 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_2^5, \mathbf{EO}_3^5), \mathbf{EW}_2^6), \mathbf{EW}_2^{6-SE}) \quad (13)$$

Next, the obtained feature maps  $\mathbf{EO}_1^6$  and  $\mathbf{EO}_2^6$  are further fused via equation (14)

$$\mathbf{EO}_1^7 = \mathbf{SGE}(\mathbf{C}(\text{concat}(\mathbf{EO}_1^6, \mathbf{EO}_2^6), \mathbf{EW}_1^7), \mathbf{EW}_1^{7-SE}) \quad (14)$$

The outputs of the  $\mathbf{SGE}$  block are then passed to an additional  $\mathbf{SGE}$  block to further downsample and ultimately achieve the bottleneck signature, the operations of which are demonstrated in equation (15).

$$\text{BottleNeck} = \mathbf{SE}(\mathbf{C}(\mathbf{EO}_1^7, \mathbf{W}_1^{\text{bottleneck}}), \mathbf{W}_1^{\text{bottleneck-SE}}) \quad (15)$$

#### ▪ Decoder

A decoder is also a network that takes the feature vectors from an encoder to map them into the intended output, segmentation images in our case, via upsampling techniques. Unlike the

convolutional layers in the encoder, which are typically designed for reducing the spatial dimensions of the input and intermediate feature maps, transposed convolutional layers are utilized for reversing the downsampling operations by the convolution.

#### *Transposed Convolutional block (TC)*

A transposed Convolutional block **TC** is composed of one transposed convolutional layer with one batch-norm layer, one ReLU activation function, and one dropout layer. A proper combination of these components is expected to accomplish the decoder's objective of converting the encoder's extracted feature maps to segmentation images. Convolutional kernels  $4 \times 4$  are defined. And all strides and paddings are set to 2 and 1, respectively.

#### *Decoder architecture*

Specifically, the decoder in our generator consists of six **TC** blocks, one output layer, and nine skip-connections. The following equations (16)-(23) illustrate the operating process of the decoder.

$$\mathbf{DEO}^1 = \text{concat}(\mathbf{TC}(\mathbf{BottleNeck}, \mathbf{DEW}^1), \mathbf{EO}_1^7) \quad (16)$$

Where  $\mathbf{DEO}^1$  and  $\mathbf{DEW}^1$  represent the output feature map and the weights of the 1<sup>st</sup> block layer in the decoder, respectively. Equation (16) depicts the fact that the bottleneck vector obtained from the encoder (15) is first passed through the first **TC** block to get the output features, then fed to the next **TC** block concatenated with  $\mathbf{EO}_1^7$  via the skip connection.

Operations similar to equation (16) for the next few layers in the decoder can be expressed as follows using equations (17)-(22):

$$\mathbf{DEO}^2 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^1, \mathbf{DEW}^2), \mathbf{EO}_1^6, \mathbf{EO}_2^6) \quad (17)$$

$$\mathbf{DEO}^3 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^2, \mathbf{DEW}^3), \mathbf{EO}_1^5, \mathbf{EO}_2^5, \mathbf{EO}_3^5) \quad (18)$$

$$\mathbf{DEO}^4 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^3, \mathbf{DEW}^4), \mathbf{EO}_1^4, \mathbf{EO}_2^4, \mathbf{EO}_3^4) \quad (19)$$

$$\mathbf{DEO}^5 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^4, \mathbf{DEW}^5), \mathbf{EO}_1^3, \mathbf{EO}_2^3, \mathbf{EO}_3^3, \mathbf{EO}_4^3, \mathbf{EO}_5^3, \mathbf{EO}_6^3) \quad (20)$$

$$\mathbf{DEO}^6 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^5, \mathbf{DEW}^6), \mathbf{EO}_1^2, \mathbf{EO}_2^2, \mathbf{EO}_3^2, \mathbf{EO}_4^2, \mathbf{EO}_5^2, \mathbf{EO}_6^2) \quad (21)$$

$$\mathbf{DEO}^7 = \text{concat}(\mathbf{TC}(\mathbf{DEO}^6, \mathbf{DEW}^7), \mathbf{EO}_1^1, \mathbf{EO}_2^1, \mathbf{EO}_3^1, \mathbf{EO}_4^1, \mathbf{EO}_5^1, \mathbf{EO}_6^1) \quad (22)$$

Unlike the previous layers, the resulting feature maps  $\mathbf{DEO}^7$  are sequentially fed into the transpose convolutional layer and Tanh activation function, to yield the segmentation image  $\mathbf{DEO}^8$ .

$$\mathbf{DEO}^8 = \mathbf{Final}(\mathbf{DEO}^7, \mathbf{DEW}^8) \quad (23)$$

Where **Final** denotes the block composed of a transposed convolutional layer and a Tanh activation function. Note that the Tanh activation function is used instead of Sigmoid in order to have the generated image fall in the range  $[-1,1]$ , as the input image is usually normalized to  $[-1,1]$ .

In short, both the above-mentioned encoder and decoder contribute to a multi-headed generator, intending to fuse the input images (six TSR images in our case) into a single unique segmentation image.

### 2.2.3 Discriminator network

Determining an image's authenticity or falsity is a critical task for discriminators in GANs [16]. In Pix2pixGAN [27], the PatchGAN approach was formulated to evaluate the local patches from the input images, which emphasizes the global structure while paying more attention to local details. However, this solution may carry the risk of losing the global features in images. Inspired by PGGAN [30], the proposed IRT-GAN model further enhances the performance of the discriminator by aggregating local and global information through a combination of PatchGAN and GlobalGAN. The architecture of the adopted discriminator in this study is shown in Fig.6.

The most frequently used network block in the discriminator  $D$  is termed **DC** block, which is composed of a convolutional layer, a BatchNorm layer, and a LeakyReLU activation function. The block in the first layer is without the BatchNorm layer. And the final layer for both PatchGAN and GlobalGAN uses the convolutional layer appended by a sigmoid function to output the scores of being real or fake.

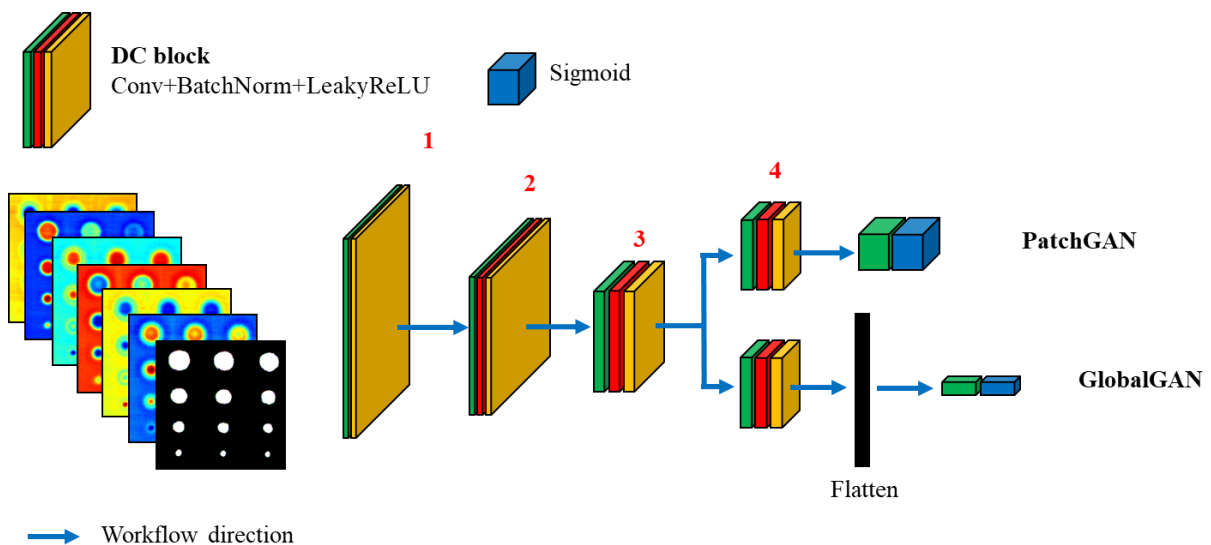


Fig.6 The architecture of the discriminator  $D$

The workflow of the discriminator can be briefly described as follows: the six TSR images

along with the ground-truth image or the segmentation image from the generator are first concatenated as input to the discriminator, passed through the first three layers, and then split into two paths: PatchGAN and GlobalGAN. In the PatchGAN path, the output feature maps from the 3<sup>rd</sup> layer will be sourced to another **DC** block and  $30 \times 30$  patch discriminators in the end. A **DC** block is appended to the output from the 3<sup>rd</sup> layer in the GlobalGAN. The resulting feature maps will be flattened and fed into a global discriminator. The mathematical expressions of the whole flow are omitted here as it is quite straightforward to understand from Fig.6.

The architectures of networks **G** and **D** in the IRT-GAN model are detailed in the Appendix, respectively.

#### 2.2.4 Objective function

During the training process, the discriminator **D** receives the generated segmented images from generator **G** or the ground-truth image along with the corresponding post-processed images (six TSR images in our case) in an attempt to have them differentiated, while the generator **G** strives to deceive the discriminator **D**. As long as **D** succeeds in correctly classifying its input, **G** benefits from the gradient provided by network **D** via its adversarial loss.

##### ▪ Adversarial loss

The adversarial loss comes from both paths of GlobalGAN and PatchGAN discriminator networks during the training process. Generator **G** and Discriminator **D** aim to tackle the targets  $\arg \min_G \max_D V_{Global\_GAN}(G, D)$  and  $\arg \min_G \max_D V_{Patch\_GAN}(G, D)$  collaboratively, which represents GlobalGAN and PatchGAN adversarial loss, respectively, as shown in equations (24) and (25).

$$L_{Global}(G, D_g) = E_{y \sim p_{data}(y)}[\log D_g(x, y)] + E_{\tilde{y} \sim p_G(\tilde{y})}[\log(1 - D_g(x, G(x)))] \quad (24)$$

$$L_{Patch}(G, D_p) = E_{y \sim p_{data}(y)}[\log D_p(x, y)] + E_{\tilde{y} \sim p_G(\tilde{y})}[\log(1 - D_p(x, G(x)))] \quad (25)$$

In which  $D_g$  and  $D_p$  denote the global and patch discriminators, respectively,  $x, y$  are the input images and corresponding ground-truth labels, and  $\tilde{y}$  is the generated image (segmentation image) from **G**. To train simultaneously for both  $D_g$  and  $D_p$ , a joint loss, aiming to capture both the holistic features in the images and the local continuity, is defined as (26)

$$L_{Joint}(G, D) = \lambda_{Global} \cdot L_{Global}(G, D_g) + \lambda_{Patch} \cdot L_{Patch}(G, D_p) \quad (26)$$

Where  $\lambda_{Global}$  and  $\lambda_{Patch}$  are the weighting parameters assigned to  $L_{Global}(G, D_g)$  and  $L_{Patch}(G, D_p)$ , respectively.

##### ▪ Reconstruction loss

The GAN model benefits from the involvement of reconstruction loss. For instance, pixel-wise

$L_1$  distance between the ground truth and generated image. The reconstruction loss  $L_{Rec}$  is defined in equation (27).

$$L_{Rec} = \frac{1}{N} \sum_{n=1}^N \frac{1}{W \cdot H \cdot C} \|y - \tilde{y}\|_1 \quad (27)$$

#### ▪ Total loss

The total loss for training the IRT-GAN can be concluded as

$$L_{Tot} = \lambda_{Rec} \cdot L_{Rec} + \lambda_{Global} \cdot L_{Global}(G, D_g) + \lambda_{Patch} \cdot L_{Patch}(G, D_p) \quad (28)$$

Note that a coefficient  $\lambda_{Rec}$  is also assigned to reconstruction loss. In this study, these parameters are set to  $\lambda_{Global} = \lambda_{Patch} = 0.5$ ,  $\lambda_{Rec} = 200$ .

### 3. Numerical and experimental datasets

#### 3.1 Numerical dataset generation for training the IRT-GAN

The demand for large datasets containing healthy samples as well as samples with diverse defect conditions makes it challenging to train deep learning models for defect detection. From the experimental side, it is not straightforward to produce a large set of samples with specific well-defined defects. Therefore, a virtual dataset has been generated through an adapted version of the IRT simulator introduced in reference [44]. In order to establish a virtual database, a square CFRP plate model of size  $150 \times 150 \times 1.84 \text{ mm}^3$  and quasi-isotropic material layup [(0/+45/90/-45)]<sub>s</sub> was adopted in this study, as shown in Fig. 7.

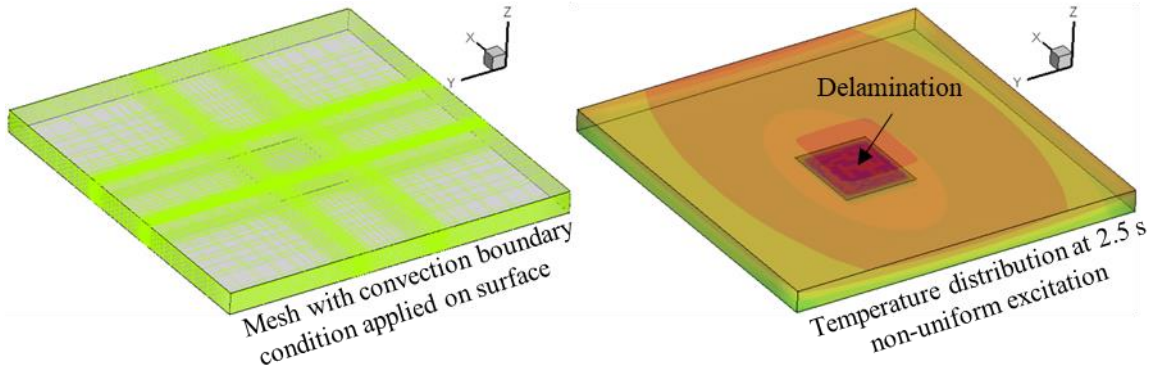


Fig.7 Numerical model of CFRP plate

There is a delamination of variable size and depth located in the middle of the part. The delamination is modelled by considering interface elements with a thermal resistance value of  $3.861e-4 \text{ K/W}$ . A mesh convergence study was performed, resulting in a discretization of approximately 65000 elements for the considered part. The size and depth of the delaminations,

as well as the material properties, are listed in Table.1. A non-uniform Gaussian-shaped heat flux with a 6 kJ is applied on the top surface of the sample, whose center is randomly located on the sample surface to mimic actual inspection conditions. The heating stage lasts 10 ms, followed by a cooling stage of 14.99 s. Convection boundary condition is considered, with a convection coefficient of 5 W/(m<sup>2</sup>·K). A total of 875 cases are simulated. After the application of a temporal standardization procedure, the TSR method (5<sup>th</sup> order polynomial) is applied to the simulated data, yielding a virtual database of  $6 \times 875 = 5250$  images.

Table.1 Parameters used in the numerical model

Quantity	Value [44]
Density $\rho$ [kg/m <sup>3</sup> ]	1530
Specific heat $c$ [kJ/(kg·K)]	917
Thermal conductivity $k$ [W/m·K]	$k_x = 2.71, k_y = 0.61, k_z = 0.53$ [45]
Side length of the delamination with rectangular shape [mm]	1.0 ~ 50.0
Depth of the delamination [mm]	0.23 ~ 1.61
The number of simulation cases	875
Simulation time	$\approx 4 \text{ min} \times 875 \text{ cases} \approx 58.3 \text{ hours}$

The obtained numerical are then fed into the IRT-GAN model for training purposes. It is also worth noting that PCT and/or PPT images may be employed as inputs for the IRT-GAN.

Due to the fact that only defects with rectangular shapes, located in the centre of the samples, have been considered in the numerical dataset, the IRT-GAN model may fall into a monolithic learning pattern and may only identify defects similar to those. In order to enrich the labelled training set and increase the diversity of defect scenarios, data augmentation techniques were applied to the numerical TSR images by leveraging input transformations that preserve output labels, as such to enable the IRT-GAN model to detect and segment regions and defect shapes that are not present in the original TSR images. The training process incorporates a variety of common and beneficial data augmentation methods, including

- ShiftScaleRotate → Randomly translate, scale, and rotate the TSR images.
- GridDistortion → Randomly distort the TSR images.
- RandomGridShuffle → Randomly shuffle grid cells on the TSR image.
- ChannelShuffle → Randomly rearrange channels of the input RGB TSR image.

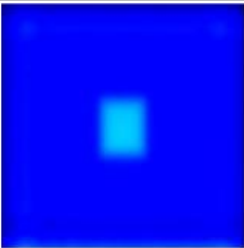

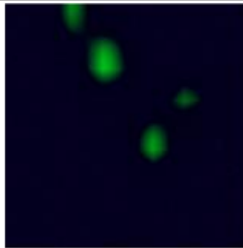

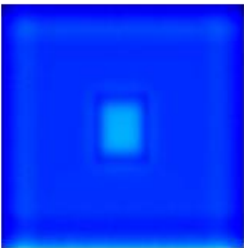

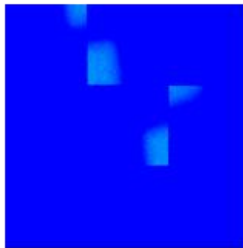

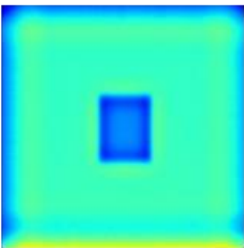



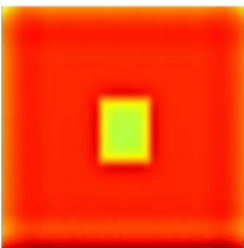



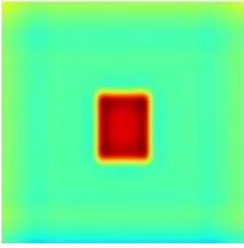

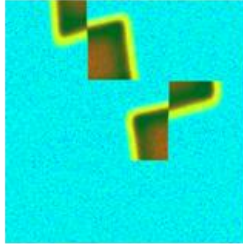
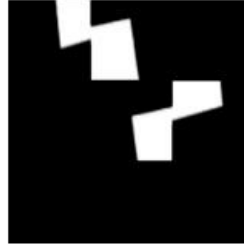
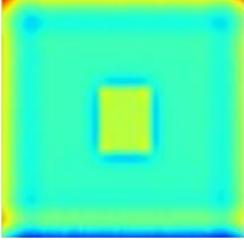
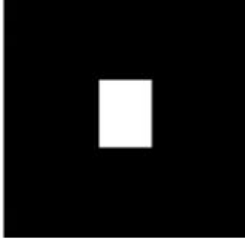
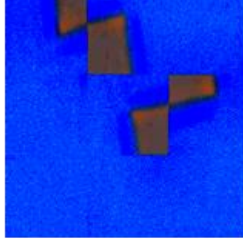
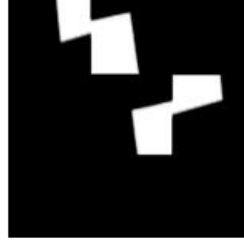
	Original TSR images	Original labels	Augmented TSR images	Augmented labels
TSR <sub>1</sub>				
TSR <sub>2</sub>				
TSR <sub>3</sub>				
TSR <sub>4</sub>				
TSR <sub>5</sub>				
TSR <sub>6</sub>				

Fig.8 Comparison between original TSR images and augmented TSR images, together with corresponding ground-truth images.

Of the augmentations discussed, nearly all of these transformations come with an associated



parameter that can be defined by users to adapt to their specific tasks. The functions employed in this study are encapsulated in PyTorch's package: Albumentations [46], which is an open-source library and a commonly used image augmentation library. Note that these functions will be sequentially implemented in the numerical TSR dataset.

Fig.8 exhibits an example of data augmentations on the obtained six TSR coefficient images from a random numerical case. The implementation of data augmentation results in an increase in diversity in terms of defect shapes, background colours, and even textures. Although the parameters associated with data augmentation for each TSR image are probably different from each other, the corresponding binary augmented labels, indicating the sizes and locations of the defects, are preserved across all TSR images.

### 3.2 Experimental dataset generation for testing the IRT-GAN

In order to evaluate the effectiveness of the proposed IRT-GAN, three experimental datasets were obtained for different composite materials, and various defect types, shapes, and depths. A Hensel linear flash lamp (6 kJ energy, 5 ms flash duration) is used to provide the optical energy input, while a FLIR A6750sc infrared camera records the sample's surface temperature in the meantime. The camera is equipped with a focal plane array of  $640 \times 512$  cryo-cooled InSb detectors and has a Noise-Equivalent Differential Temperature (NEDT) of  $\leq 20$  mK and a bit depth of 14 bits. It is sensitive within the mid-infrared wavelength range of  $3-5 \mu\text{m}$ . A 50mm lens is mounted on the camera, and the inspected samples are positioned at a distance such that they fill the field of view of the camera. Hard- and software modules from edevis GmbH ensure accurate synchronization between the optical excitation and the data acquisition. A schematic of the thermographic inspection setup is shown in Fig.9(a).

The first inspected sample is an autoclave manufactured cross-ply  $[(0/90)_5]_s$  GFRP laminate with a total thickness of 5.0 mm, and a set of circular Flat Bottom Holes (FBHs) has been milled from the backside. In total, 32 FBH's were created with diameters of 5, 10, 15, and 20 mm, and with remaining thicknesses ranging from 0.5 mm up to 4.0 mm (in steps of 0.5 mm). A photograph of the sample, which will be called  $\text{GFRP}_{\text{FBH}}$  further on, is provided in Fig.9(b). The distance between the GFRP sample and the flash optical excitation was 500 mm. The distance between the IR camera and the GFRP sample was 1050 mm, and the cooling regime was recorded for 120 s at a framerate of 30 Hz.

Next, an autoclave manufactured CFRP plate with a thickness of 5.5 mm and a quasi-isotropic layup of  $[(+45/0/-45/90)_3]_s$  is studied. 12 circular FBHs were milled from the backside of the sample, with diameters ranging from 7 mm up to 25 mm and a remaining thickness of up to

3.97 mm. Note that FBH 1 completely penetrates the part, and as such, is a through-hole. A photograph of the sample, later on, referred to as CFRP<sub>FBH</sub>, and the defect parameters are displayed in Fig.9(c). The excitation lamp was placed at an offset of 400 mm, while the sample's temperature response was recorded for 120 s at 30 Hz at a distance of 1050 mm.

The third sample of interest in this research is a CFRP laminate with a thickness of 3 mm and a layup of  $[(0/90)_3]_s$ . The sample was manufactured using the resin transfer moulding procedure, and a range of inter-ply square inserts was introduced during production. The inserts are constructed from double-folded brass foil encapsulated with flash breaker tape and measure  $15 \times 15 \text{ mm}^2$  and  $20 \times 20 \text{ mm}^2$ . They are located at depths ranging from 0.25 mm to 1.5 mm on the back of the plate opposite the heat source. At the resin injection point in the middle of the CFRP plate, an additional unintended defect was introduced during the manufacturing. A photograph and the information of this sample, named as CFRP<sub>Insert</sub>, are provided in Fig.9(d). The sample was located at a distance of 400 mm from the flash lamp, and at 1000 mm from the IR camera (which recorded the cooling down regime for 40 s at 30 Hz).

The last sample is an autoclave manufactured CFRP plate with a thickness of 5.5 mm and a quasi-isotropic layup of  $[(+45/0/-45/90)_3]_s$  which was subjected to a low-energy impact event according to ASTM standard D7136 [48]. The impactor with a weight of 7.72 kg is equipped with an Endevco Isotron 23-1 load cell holding a 16 mm diameter hemispherical hardened solid steel impact tip. It was dropped from a height of 0.1 m on this sample, and resulting in a measured impact energy of 6 J. This impact event leads to the formation of barely visible impact damage in the CFRP laminate [ref to <https://doi.org/10.1016/j.polymertesting.2017.11.023>]. A 5 MHz ultrasonic C-scan in transmission has been performed in order to assess the size of the impact-induced damage and to obtain a ground truth image of the damage. Fig.9(e) shows a photograph of this sample (C-scan data is overlaid), which will be referred to as CFRP<sub>Impact</sub>. The distance between the CFRP<sub>Impact</sub> sample and the flash optical excitation was 500 mm. The distance between the IR camera and the CFRP<sub>Impact</sub> sample was 1050 mm, and the cooling regime was recorded for 120 s at a framerate of 30 Hz.

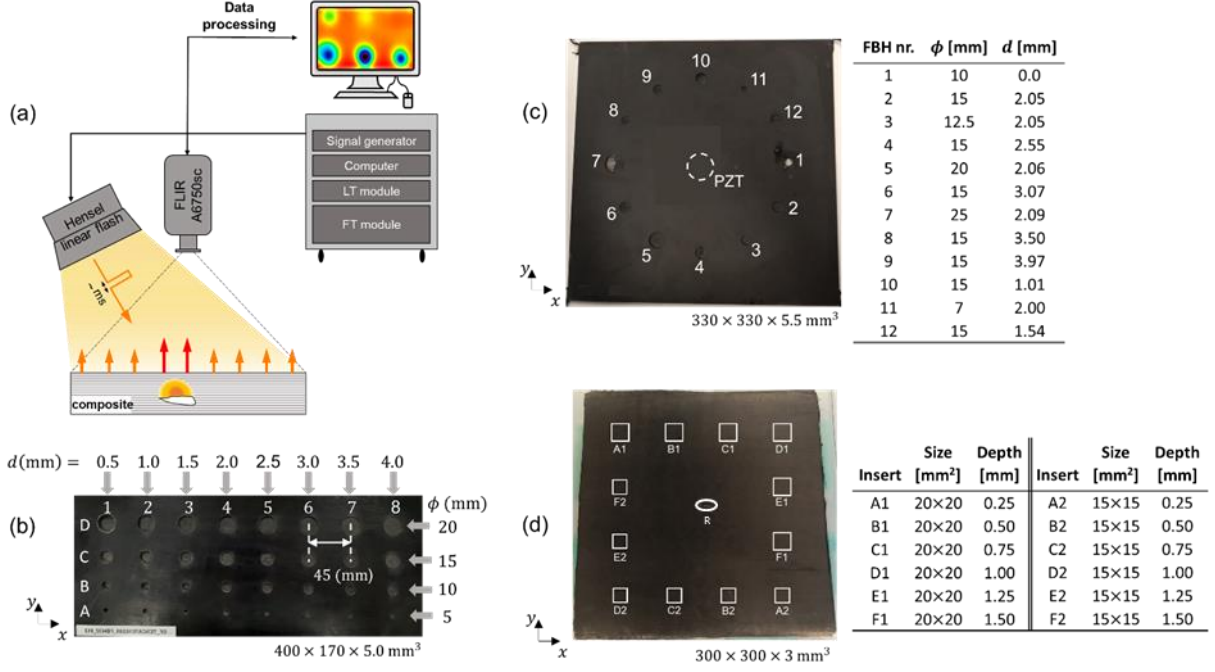


Fig.9: (a) Schematic of experimental setup for flash thermography, (b-e) photographs and defect parameters of GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, CFRP<sub>Insert</sub> and CFRP<sub>Impact</sub>.

### 3.3 Evaluation metrics

In this study, the Pixel Accuracy (PA) and Intersection over Union (IoU) values, as the frequently-used evaluation metrics to measure the segmentation performance on a specific dataset [47], were adopted to validate the effectiveness of the proposed IRT-GAN model.

PA computes a ratio between the amount of properly classified pixels and its total number, and its definition is formulated as follows:

$$PA = \frac{\sum_{i=0}^k p_{ii}}{\sum_{i=0}^k \sum_{j=0}^k p_{ij}} \quad (29)$$

In which  $k$  and  $p_{ij}$  denote the total classes and the number of pixels of class  $i$  inferred to class  $j$ , respectively. In our study,  $k=0$  and  $k=1$  indicate the healthy and defective areas, respectively. And  $p_{ii}$  and  $p_{jj}$  here stand for the number of true positives and negatives, while  $p_{ij}$  and  $p_{ji}$  are usually interpreted as false positives and false negatives.

The IoU value is defined as

$$IoU = \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k p_{ji} - p_{ii}} \quad (30)$$

and measure whether the target in the image is detected.

## 4. Results and analysis

### 4.1 Training details

The IRT-GAN is implemented using the PyTorch framework in Python (Python 3.7.10 and CUDA v11.0.221) using an NVIDIA Tesla P100 GPU card with 12GB of RAM. The IRT-GAN model employs the Adam optimizer with an initial learning rate  $lr = 1e^{-4}$ , and momentum parameters  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ . The batch size for training is 16. The equality of  $\lambda_{Global}$  and  $\lambda_{Patch}$  enables the discriminator to capture both local and global information. The training epoch is set to be 100, empirically shown to yield a good performance.

To demonstrate the performance of the proposed IRT-GAN model, we train the model on the augmented TSR images from the virtual dataset and then validate the trained model on the three experimental datasets ( $GFRP_{FBH}$ ,  $CFRP_{FBH}$ , and  $CFRP_{Insert}$ ). Prior to the training process, all TSR images have to be resized to  $256 \times 256$  (pixel $\times$ pixel) in order to comply with the input requirement of the proposed IRT-GAN model.

### 4.2 Experimental results

The qualitative evaluation of the proposed IRT-GAN model is reported in Fig.10 for the different inspected composite samples.

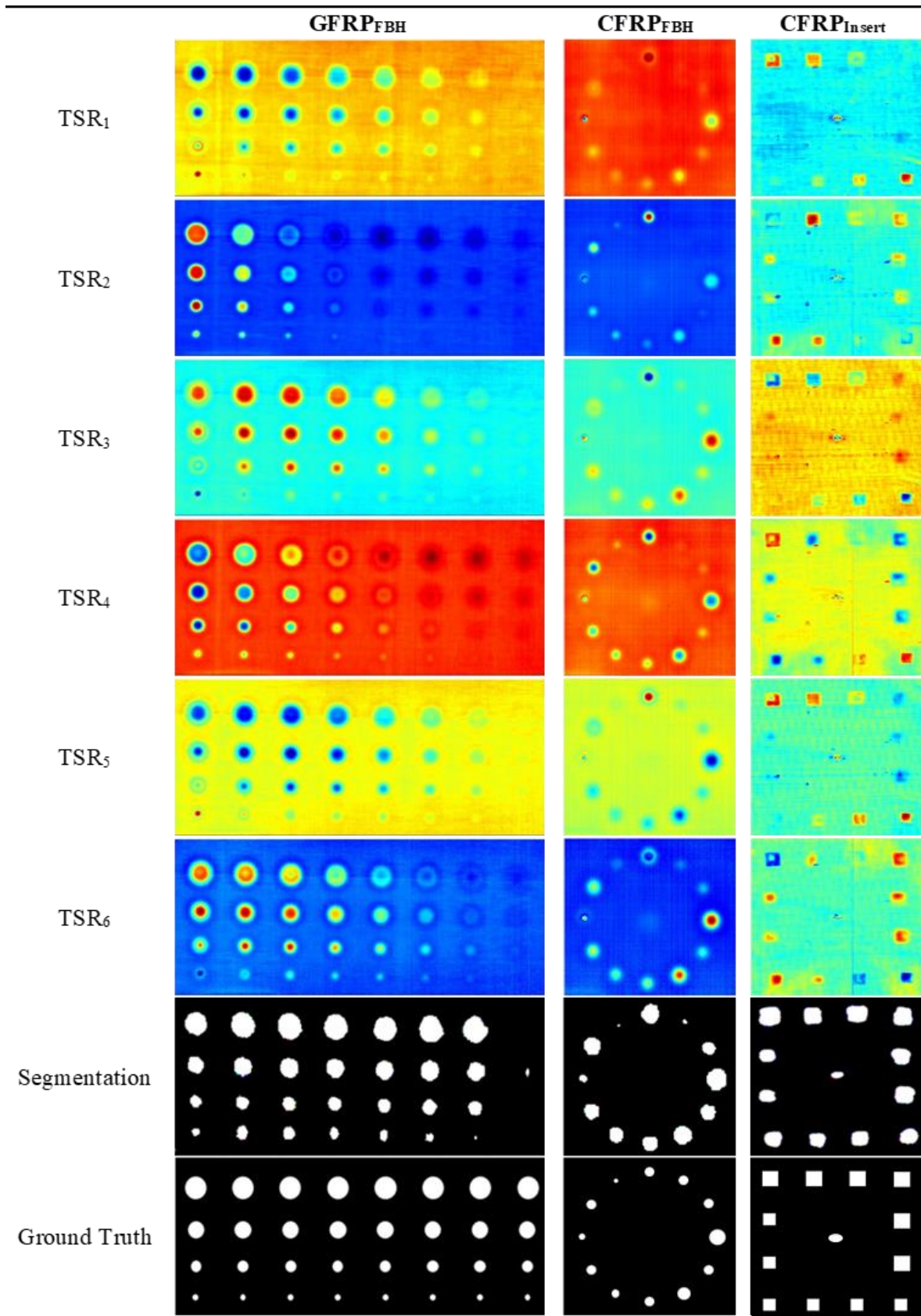


Fig.10 Visualization of each TSR image, predicted segmentation, and its ground truth for the  $GFRP_{FBH}$ ,  $CFRP_{FBH}$ ,  $CFRP_{Insert}$  and  $CFRP_{Impact}$  samples

$GFRP_{FBH}$	$CFRP_{FBH}$	$CFRP_{Insert}$
--------------	--------------	-----------------

PA	93.79%	96.20%	91.22%
IoU	56.77%	44.93%	42.97%

Table.2 PA and IoU for the GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples

Although the IRT-GAN is trained exclusively on numerical data, it correctly identifies the majority of defects in the three GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples, demonstrating the good performance of the proposed fusion strategy in the encoder to learn the cross-correlation between the inputs and to enhance the feature extraction and anti-noise interference capability. More precisely, a detectability rate of 29/32 was obtained for the GFRP<sub>FBH</sub> sample. Three of the deepest defects were not successfully detected by the IRT-GAN. For the CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples, all defects have been successfully identified. It is also worth noting that the IRT-GAN appears to be quite accurate at predicting the defect sizes. Though, for sample CFRP<sub>Impact</sub>, IRT-GAN clearly underestimates the size of the damage. This can be easily understood by considering the diffusive nature of heat waves (with high lateral diffusion in case of CFRP), combined with the fact that the impact damage extends through the whole depth of the laminate. Hence, the small defect fragments at large depth are not properly represented in the acquired thermal data, and as such can also not be detected by the IRT-GAN framework.

The obtained PA and IoU values for the GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples are summarized in Table.2. The low IoU values may be due to an imperfect match between the actual defect and the ground truth image. These results suggest that the IRT-GAN is well-suited to performing defect segmentation tasks for IRT-based images by fusing TSR images. Note that the structure can be easily applied to PPT, PCT, or other pre-processed images.

#### 4.3 Comparison with other architectures

This section investigates the influence of the fusion stage on the proposed multi-headed IRT-GAN. In the implementation described above, the fusion is performed in the middle of the network. However, the fusion process can also be used at the beginning or end of the network. We refer to this as IRT-GAN\_*Middle*, IRT-GAN\_*Early*, and IRT-GAN\_*Late* in the following content. Only a slight modification was made to the architecture of the generator  $G$  to accommodate the various fusion versions, and all three IRT-GAN models employ the identical discriminator  $D$  shown in Fig.6. Additionally, both  $SGE$  and  $C$  blocks are preserved to ensure a meaningful comparison.

Fig.11 demonstrates the architecture of IRT-GAN\_*Early*. Rather than fusing feature maps in the intermediate layers of the encoder, the six TSR images are directly concatenated channel-wise

at the beginning to form a new matrix that serves as the input to the  $G$  network without any preprocessing.

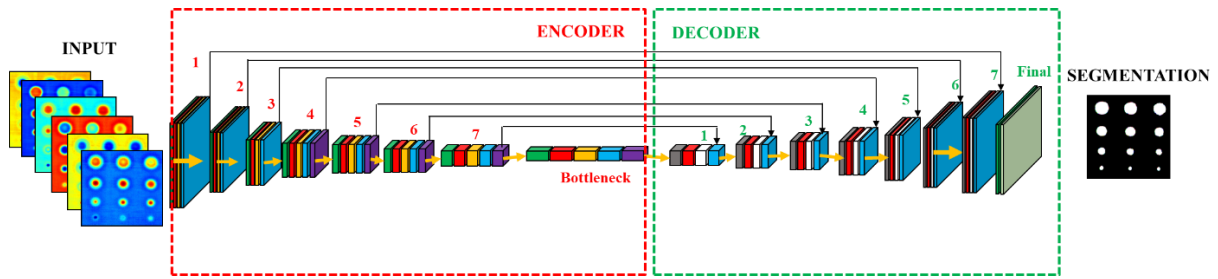


Fig.11 The architecture of the generator  $G$  in IRT-GAN\_Early

Following this, the new input progressively learns a joint representation for the six initial TSR images, eventually obtaining a unified representation. It should be noted that, while the early fusion method is simple and straightforward, it runs the risk of sacrificing the proper extraction of independent features from each TSR image early in the process, resulting in the interfusion of irrelevant features and a decrease in fusion power.

Another extreme variant is the IRT-GAN\_Late, in which the multi-image fusion is done at the end of the network, see Fig.12. IRT-GAN\_Late simply merges the outcome from each  $G$  stream by max-pooling vote. This severely restricts the potential to exploit the cross-correlation between different single images.

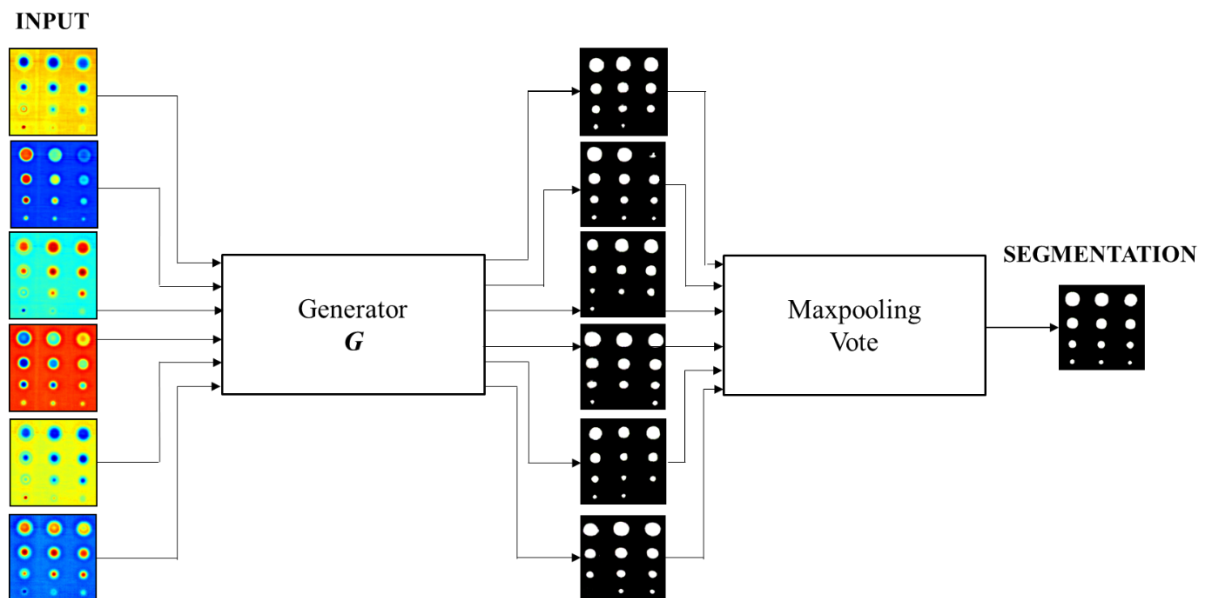


Fig.12 The architecture of the generator  $G$  in IRT-GAN\_Late

The training process of generator  $G$  within the late fusion requires One-to-One image translation, i.e. each numerical TSR image corresponds to one ground-truth label, instead of Multi-to-One in IRT-GAN\_Middle and IRT-GAN\_Early. After the training, each TSR image serves as an independent input to achieve the segmentation result via the well-trained  $G$ . Next,

following the max-pooling voting mechanism at the pixel level, the segmentation prediction can be eventually obtained. The max-pooling voting mechanism employed here is presented in equation (31).

$$\begin{aligned}
 y(w, h, C) &= \max\{y_i(w, h, C)\} \\
 i &= 1, 2, \dots, 6; \\
 0 &\leq w \leq W; \\
 0 &\leq h \leq H
 \end{aligned}
 \tag{31}$$

In which  $y_i$  stands for the segmentation result for the  $i$ th TSR image. The channel number for the binary segmentation map is set to  $C = 1$ .

While the adopted voting mechanism may improve defect detection, it also increases the risk of misjudgment, in which defect-free regions are incorrectly identified as defects. And this is largely a direct result of the late fusion strategy's failure to take advantage of cross-correlation between individual TSR images.

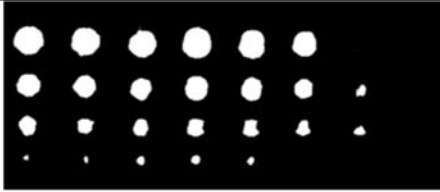
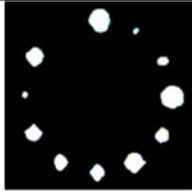
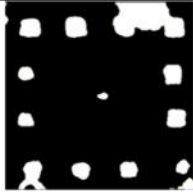
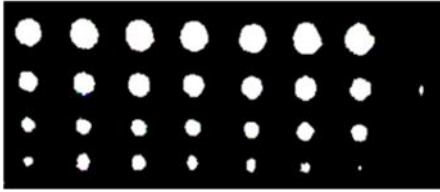
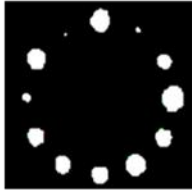
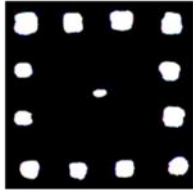
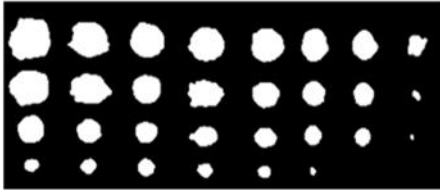
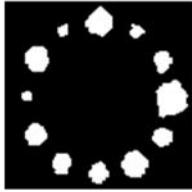

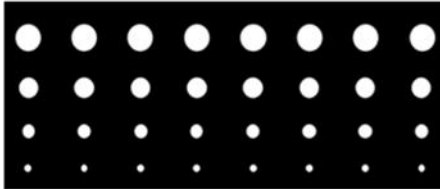
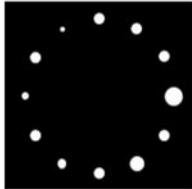
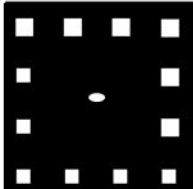
Model	GFRP <sub>FBH</sub>	CFRP <sub>FBH</sub>	CFRP <sub>Insert</sub>
IRT-GAN <i>Early</i>	 25/32 detected	 11/12 detected	 12/12 detected
IRT-GAN <i>Middle</i>	 29/32 detected	 12/12 detected	 12/12 detected
IRT-GAN <i>Late</i>	 30/32 detected	 12/12 detected	 12/12 detected
Ground Truth			

Fig.13 Visualization of predicted segmentations for the GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples at three different fusion stages

In section 4.3, we trained and evaluated the IRT-GAN\_Early, IRT-GAN\_Middle, and IRT-



GAN\_Late models on the same virtual and experimental datasets in order to evaluate their performance in defect detection. Fig.13 summarizes the predicted segmentation maps for the GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples at early, middle and late fusion stages. IRT-GAN\_Early has the lowest defect detectability and misses several defects. The IRT-GAN\_Late model, on the other hand, has the highest defect detectability, but it enlarges and distorts the defect shapes significantly, and even highlights background noise and artifacts. From these results, it seems that the IRT-GAN\_Middle shows the highest performance by keeping a balance between defect detectability, shape reconstruction, and background suppression.

Table.3 presents quantitative results where the PA and IoU values for each sample are reported, as well as D-PA and H-PA indicators representing the defective pixel accuracy and healthy pixel accuracy, respectively. The results indicate that IRT-GAN\_Middle is superior to the other fusion variants, with high PA and IoU values in each of the three different composite samples. The reason for this is that the multi-headed IRT-GAN\_Middle can make better use of the information from multiple TSR images due to its progressive fusion strategy that tends to learn independent feature maps for each TSR image in the initial layers and then integrates or fuses them progressively to a unique feature representative until the layer before the bottleneck layer in the encoder. Notably, the enlargement and distortion of the defect segmentation using the IRT-GAN\_Late model lead to the highest D-PA values.

Model	GFRP <sub>FBH</sub>		CFRP <sub>FBH</sub>		CFRP <sub>Insert</sub>	
IRT-GAN_Middle	PA	<b>93.79%</b>	PA	<b>96.20%</b>	PA	<b>91.22%</b>
	D-PA	78.00%	D-PA	87.19%	D-PA	70.16%
	H-PA	<b>95.17%</b>	H-PA	<b>96.53%</b>	H-PA	<b>93.39%</b>
	IoU	<b>56.77%</b>	IoU	<b>44.93%</b>	IoU	<b>42.97%</b>
IRT-GAN_Early	PA	91.84%	PA	96.02%	PA	87.82%
	D-PA	64.16%	D-PA	83.89%	D-PA	66.92%
	H-PA	94.84%	H-PA	96.47%	H-PA	89.98%
	IoU	47.39%	IoU	42.97%	IoU	33.96%
IRT-GAN_Late	PA	86.41%	PA	93.65%	PA	78.34%
	D-PA	<b>81.29%</b>	D-PA	<b>90.37%</b>	D-PA	<b>85.00%</b>
	H-PA	87.12%	H-PA	93.81%	H-PA	77.67%
	IoU	42.14%	IoU	39.29%	IoU	26.84%

Table.3 The PA and IoU results of predicted segmentations for the GFRP<sub>FBH</sub>, CFRP<sub>FBH</sub>, and CFRP<sub>Insert</sub> samples at three different fusion stages

## 5. Conclusions

In this paper, we introduced a multi-headed IRT-GAN model, a novel architecture to perform defect segmentation tasks from IRT images via a progressive fusion strategy. The IRT-GAN

takes a set of images as input, six TSR images in our case, and progressively integrates the information contained in each TSR image via a dedicated multi-headed encoder. These learned feature maps in the encoder are then propagated to the corresponding decoding layers via skip-connections. After the adversarial training, a defect segmentation map is achieved. Additionally, SGE blocks for enhancing semantic feature learning are engaged in the feature fusion process in the encoder. This approach enables the IRT-GAN to be trained on an augmented virtual dataset, and then applied to experimental data.

Results are presented for three composite samples with a range of defect types, sizes, and depths. It is demonstrated qualitatively and quantitatively that the multi-headed IRT-GAN model achieves high performance in segmenting the experimental datasets. The influence of the fusion strategies, i.e. early, middle, and late fusion in the generator network, has also been discussed. The results show the benefit of middle-stage fusion in enhancing the segmentation performance. Future work will focus on the application of the proposed IRT-GAN model for assessing complex defects in components with industrial complexity, and on the estimation of defect depth.

## 6. CRediT authorship contribution statement

**Liangliang Cheng**: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Zongfei Tong**: Numerical dataset creation, Writing - review & editing. **Shejuan Xie**: Writing - review & editing. **Mathias Kersemans**: Conceptualization, Writing - review & editing, Supervision, Project administration.

## 7. Acknowledgments

This work has been funded by Bijzonder Onderzoeksfonds Ghent University (BOF 01N01719). Zongfei Tong is a joint Ph.D. student between Ghent University and Xi'an Jiaotong University and is supported by the China Scholarship Council by grant number 202006280481.

## 8. Appendix

### 8.1 Generator architecture

Let conv., BN, LR, SGE denote convolutional layer, batch norm layer, LeakyRelu layer, and Spatial Group-wise Enhance layer, respectively, in the encoder of generator network.

The encoder architecture of the proposed IRT-GAN model consists of nine layers, shown in

Fig.14.

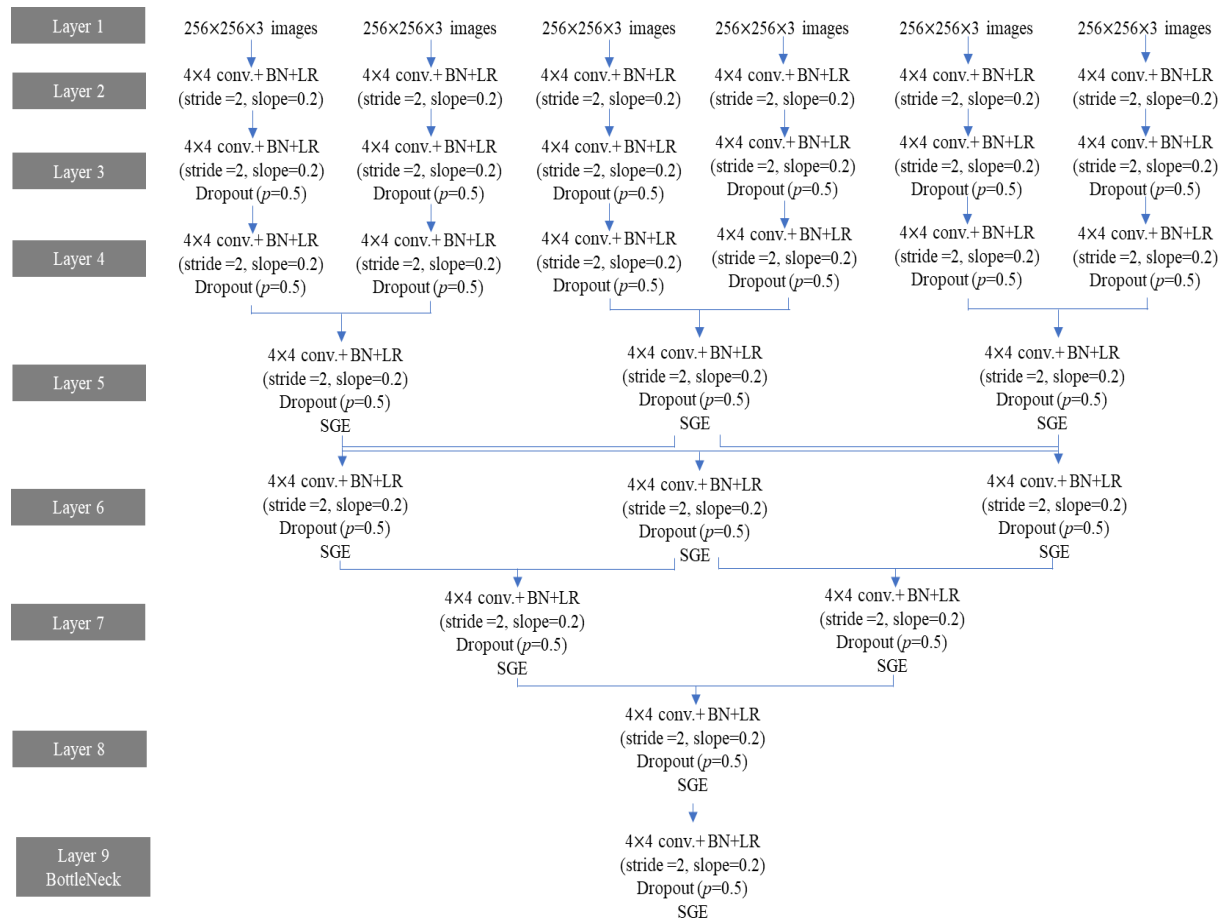


Fig.14 The architecture of encoder in the IRT-GAN model

Let convTranspose and RL denote transposed convolutional layer, and ReLu layer, respectively, in the decoder of the generator network.

The decoder architecture of the proposed IRT-GAN model consists of eight layers, shown in Fig.15.

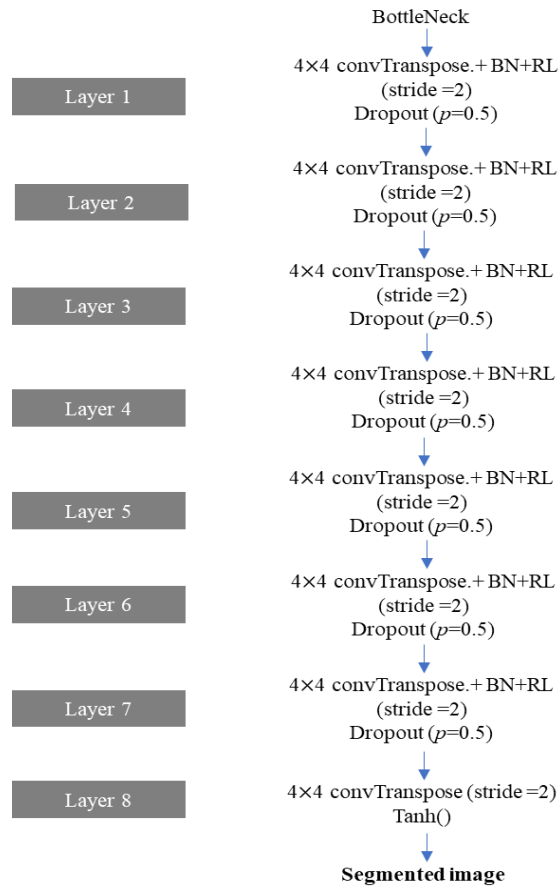


Fig.15 The architecture of decoder in the IRT-GAN model

## 8.2 Discriminator architecture

The discriminator architecture of the proposed IRT-GAN model consists of seven layers, shown in Fig.16.

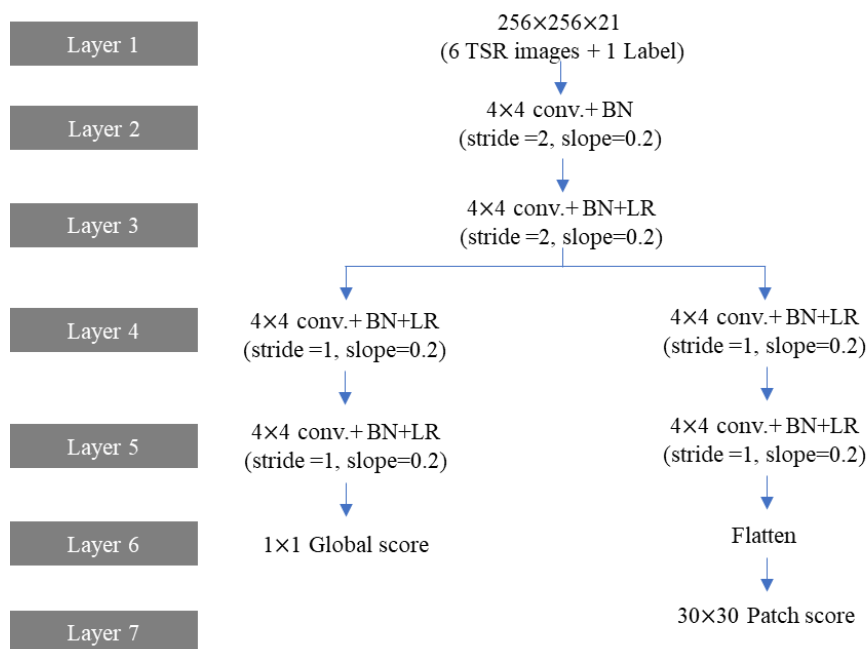


Fig.16 The architecture of discriminator in the IRT-GAN model

## 9. References

1. Ciampa, F., Mahmoodi, P., Pinto, F., & Meo, M. (2018). Recent advances in active infrared thermography for non-destructive testing of aerospace components. *Sensors*, 18(2), 609.
2. Ibarra-Castanedo, C., Genest, M., Piau, J. M., Guibert, S., Bendada, A., & Maldague, X. P. (2007). Active infrared thermography techniques for the nondestructive testing of materials. In *Ultrasonic and advanced methods for nondestructive testing and material characterization* (pp. 325-348).
3. Li, Y., Yang, Z. W., Zhu, J. T., Ming, A. B., Zhang, W., & Zhang, J. Y. (2016). Investigation on the damage evolution in the impacted composite material based on active infrared thermography. *Ndt & E International*, 83, 114-122.
4. Oswald-Tranta, B., & Shepard, S. M. (2013, May). Comparison of pulse phase and thermographic signal reconstruction processing methods. In *Thermosense: Thermal Infrared Applications XXXV* (Vol. 8705, p. 87050S). International Society for Optics and Photonics.
5. Balageas, D. L., Roche, J. M., Leroy, F. H., Liu, W. M., & Gorbach, A. M. (2015). The thermographic signal reconstruction method: a powerful tool for the enhancement of transient thermographic images. *Biocybernetics and biomedical engineering*, 35(1), 1-9.
6. Shepard, S. M., & Beemer, M. F. (2015, May). Advances in thermographic signal reconstruction. In *Thermosense: thermal infrared applications XXXVII* (Vol. 9485, p. 94850R). International Society for Optics and Photonics.
7. Rajic, N. (2002). Principal component thermography for flaw contrast enhancement and flaw depth characterisation in composite structures. *Composite structures*, 58(4), 521-528.
8. Winfree, W. P., Cramer, K. E., Zalameda, J. N., Howell, P. A., & Burke, E. R. (2015, May). Principal component analysis of thermographic data. In *Thermosense: Thermal Infrared Applications XXXVII* (Vol. 9485, p. 94850S). International Society for Optics and Photonics.
9. Fleuret, Julien R., Samira Ebrahimi, Clemente Ibarra-Castanedo, and Xavier PV Maldague. "Independent Component Analysis Applied on Pulsed Thermographic Data for Carbon Fiber Reinforced Plastic Inspection: A Comparative Study." *Applied Sciences* 11, no. 10 (2021): 4377.
10. Liu, Yi, Jin-Yi Wu, Kaixin Liu, Hsiu-Li Wen, Yuan Yao, Stefano Sfarra, and Chunhui Zhao. "Independent component thermography for non-destructive testing of defects in polymer composites." *Measurement Science and Technology* 30, no. 4 (2019): 044006.

11. Maldague, X., & Marinetti, S. (1996). Pulse phase infrared thermography. *Journal of applied physics*, 79(5), 2694-2698.
12. Maldague, X., Galmiche, F., & Ziadi, A. (2002). Advances in pulsed phase thermography. *Infrared physics & technology*, 43(3-5), 175-181.
13. Balageas, D. L., Roche, J. M., Leroy, F. H., Liu, W. M., & Gorbach, A. M. (2015). The thermographic signal reconstruction method: a powerful tool for the enhancement of transient thermographic images. *Biocybernetics and biomedical engineering*, 35(1), 1-9.
14. Chung, Y., Shrestha, R., Lee, S., & Kim, W. (2020). Thermographic Inspection of Internal Defects in Steel Structures: Analysis of Signal Processing Techniques in Pulsed Thermography. *Sensors*, 20(21), 6015.
15. Zheng, K., Chang, Y. S., Wang, K. H., & Yao, Y. (2015). Improved non-destructive testing of carbon fiber reinforced polymer (CFRP) composites using pulsed thermograph. *Polymer Testing*, 46, 26-32.
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.
18. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1), 53-65.
19. Gong, Y., Shao, H., Luo, J., & Li, Z. (2020). A deep transfer learning model for inclusion defect detection of aeronautics composite materials. *Composite Structures*, 252, 112681.
20. Bang, H. T., Park, S., & Jeon, H. (2020). Defect identification in composite materials via thermography and deep learning techniques. *Composite Structures*, 246, 112405.
21. Ruan, L., Gao, B., Wu, S., & Woo, W. L. (2020). DefectNet: Joint loss structured deep adversarial network for thermography defect detecting system. *Neurocomputing*, 417, 441-457.
22. Luo, Q., Gao, B., Woo, W. L., & Yang, Y. (2019). Temporal and spatial deep learning network for infrared thermal defect detection. *NDT & E International*, 108, 102164.
23. C. Maierhofer, P. Myrach, M. Reischel, H. Steinfurth, M. Röllig and M. Kunert (2014). "Characterizing damage in CFRP structures using flash thermography in reflection and transmission configurations." *Composites Part B: Engineering* 57: 35-46.

24. D. P. Almond, S. L. Angioni and S. G. Pickering (2017). "Long pulse excitation thermographic non-destructive evaluation." *NDT & E International* 87: 7-14.
25. D'Accardi E, Palumbo D, Galietti U, (2021), A comparison among different ways to investigate composite materials with lock-in thermography: The multi-frequency approach. *Materials*, 14 (10), 2525.
26. S. Hedayatrasa, G. Poelman, J. Segers, W. Van Paepegem and M. Kersemans (2021). "On the application of an optimized Frequency-Phase Modulated waveform for enhanced infrared thermal wave radar imaging of composites." *Optics and Lasers in Engineering* 138: 106411.
27. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
28. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., ... & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops* (pp. 0-0).
29. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).
30. Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*.
31. Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., & Liao, J. (2021). PD-GAN: Probabilistic Diverse GAN for Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9371-9381).
32. Springenberg, J. T. (2015). Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*.
33. Liu, H., Zhou, J., Xu, Y., Zheng, Y., Peng, X., & Jiang, W. (2018). Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. *Neurocomputing*, 315, 412-424.
34. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
35. Du, G., Cao, X., Liang, J., Chen, X., & Zhan, Y. (2020). Medical image segmentation based on u-net: A review. *Journal of Imaging Science and Technology*, 64(2), 20508-1.

36. Siddique, N., Sidike, P., Elkin, C., & Devabhaktuni, V. (2020). U-Net and its variants for medical image segmentation: theory and applications. arXiv preprint arXiv:2011.01118.
37. Li, X., Hu, X., & Yang, J. (2019). Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. arXiv preprint arXiv:1905.09646.
38. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
39. Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.
40. Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).
41. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
42. Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010, June). Deconvolutional networks. In 2010 IEEE Computer Society Conference on computer vision and pattern recognition (pp. 2528-2535). IEEE.
43. Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).
44. H. Liu, S. Xie, C. Pei, J. Qiu, Y. Li, Z. Chen, Development of a Fast Numerical Simulator for Infrared Thermography Testing Signals of Delamination Defect in a Multilayered Plate, *Ieee Transactions on Industrial Informatics*, 14 (2018) 5544-5552.
45. C. Maierhofer, M. Röllig, M. Gower, M. Lodeiro, G. Baker, C. Monte, A. Adibekyan, B. Gutschwager, L. Knazowicka, A. Blahut, Evaluation of different techniques of active thermography for quantification of artificial defects in fiber-reinforced composites using thermal and phase contrast data analysis, *International Journal of Thermophysics*, 39 (2018) 1-37.
46. Buslaev, Alexander, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. "Albumentations: fast and flexible image augmentations." *Information* 11, no. 2 (2020): 125.
47. Zhao, W., Fu, Y., Wei, X., & Wang, H. (2018). An improved image semantic segmentation method based on superpixels and conditional random fields. *Applied Sciences*, 8(5), 837.
48. Standard practice for ultrasonic testing of flat panel composites and sandwich core materials used in aerospace applications. West Conshohocken, PA: ASTM International; 2012.



