



# An unsupervised defect detection model for a dry carbon fiber textile

Martin Szarski<sup>1</sup> · Sunita Chauhan<sup>1</sup>

Received: 10 February 2021 / Accepted: 4 May 2022 / Published online: 6 June 2022  
© The Author(s) 2022

## Abstract

Inspection of dry carbon textiles is a key step to ensure quality in aerospace manufacturing. Due to the rarity and variety of defects, collecting a comprehensive defect dataset is difficult, while collecting ‘normal’ data is comparatively easy. In this paper, we present an unsupervised defect detection method for carbon fiber textiles that meets four key criteria for industrial applicability: using only ‘normal’ data, achieving high accuracy even on small and subtle defects, allowing visual interpretation, and achieving real-time performance. We combine a Visual Transformer Encoder and a Normalizing Flow to gather global context from input images and directly produce an image likelihood which is then used as an anomaly score. We demonstrate that when trained on only 150 normal samples, our method correctly detects 100% of anomalies with a 0% false positive rate on a industrial carbon fabric dataset with 34 real defect samples, including subtle stray fiber defects covering only 1% image area where previous methods are shown to fail. We validate the performance on the large public defect dataset *MVTec-AD Textures*, where we outperform previous work by 4–10%, proving the applicability of our method to other domains. Additionally, we propose a method to extract interpretable anomaly maps from Visual Transformer Attention Rollout and Image Likelihood Gradients that produces convincing explanations for detected anomalies. Finally, we show that the inference time for the model is acceptable at 32 ms, achieving real-time performance.

**Keywords** Quality control · Defect detection · Vision inspection · Anomaly detection · Unsupervised learning

## Introduction

The climate-change imperative for more fuel-efficient airliners has increased demand for more efficient structures and materials. Subsequently, the use of structures built with carbon fiber reinforced plastic (CFRP) has become essential in the production of modern airliners, and seen an exponential increase since its introduction (Angerer et al. 2010) (see Fig. 1).

A notional process flow for Vacuum Assisted Resin Transfer Moulding (VARTM) is displayed in the following block diagram (Fig. 2), adapted from the aerospace industry case studies documented by Björnsson et al. (2015).

Dry fiber textiles are provided on rolls of fabric which are then placed on to an automated cutting machine that cuts out individual ply shapes from CAD data. Then the individual plies are manually handled, stacked, before being manually placed and draped one by one on the layup tool (mandrel).

The tool and dry carbon preform are then vacuum bagged, the resin is heated and injected, and then heat is applied to cure the part into its final state. While the final cured part can be inspected with traditional non-destructive techniques (e.g. thermography, ultrasound), for infusion based processes it is important to control quality before the resin infusion step (Heuer et al. 2015).

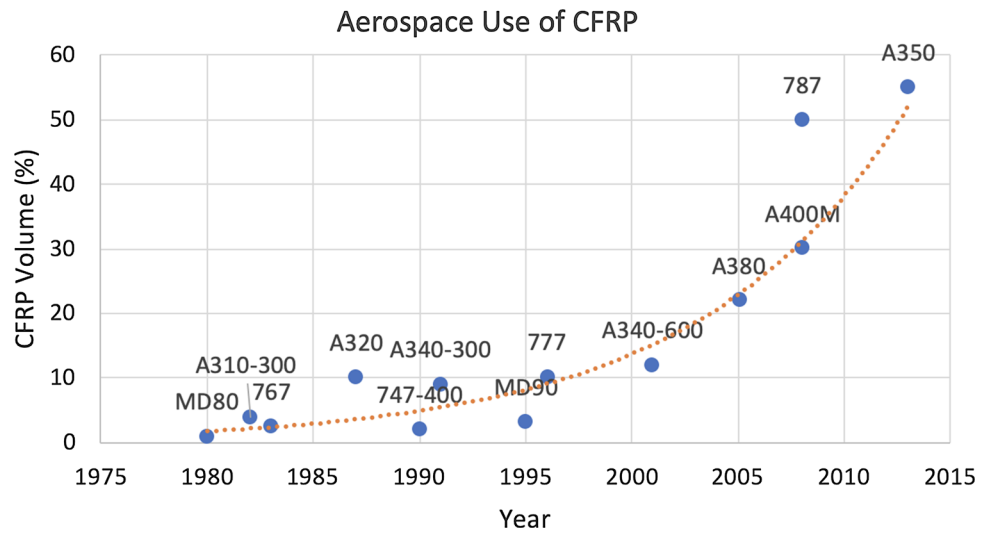
The assembly of dry carbon fiber textile sheets (plies) into a preform typically involves manual steps with no integrated quality control such as preform assembly. In the aerospace industry with a high variety of complex parts, the lack of automation inhibits the ability to achieve the required production rates in a cost-effective manner (Angerer et al. 2010). Due to limitations in automated machines involved in textile manufacture, cutting, handling, and layup, a fully automated inspection system capable of high-rates is required to enable a fully automated dry CFRP production system, as the full benefit of automated handling is only captured when the process does not need to halt for time consuming manual inspection.

For an overview of various approaches to automated dry fiber fabric layup we refer readers to work by Elkington et al. (2017) and for a comprehensive review covering both

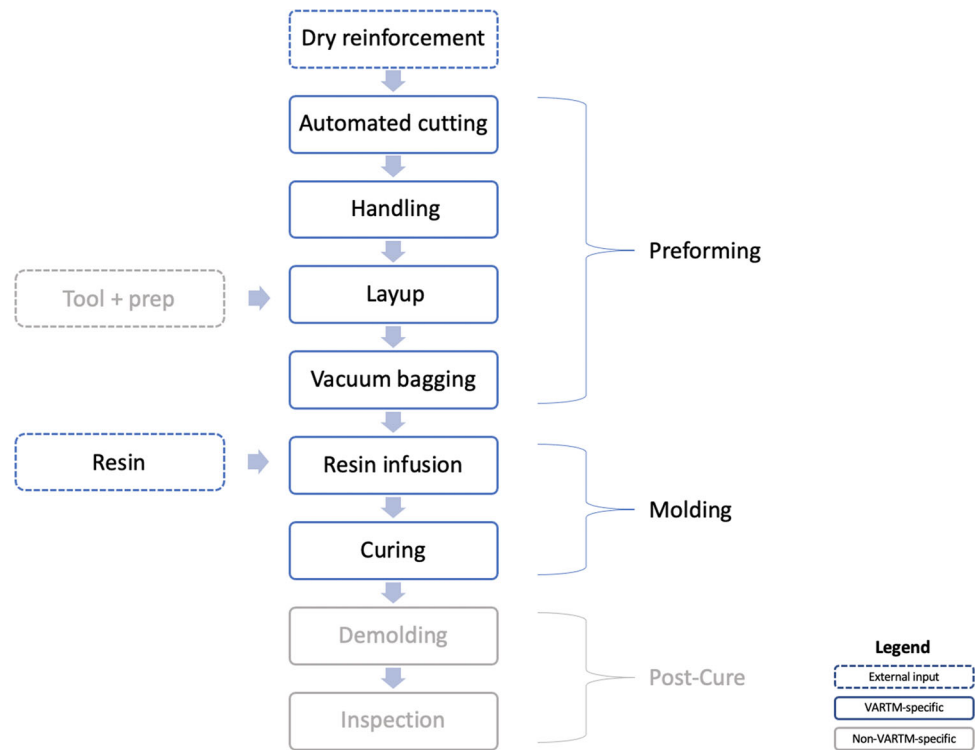
✉ Sunita Chauhan  
sunita.chauhan@monash.edu

<sup>1</sup> Department of Mechanical and Aerospace Engineering,  
Monash University, Melbourne, Australia

**Fig. 1** Aerospace CFRP use data from Angerer et al. (2010)



**Fig. 2** VARTM Process Steps adapted from Björnsson et al. (2015)



dry and prepreg handling please see the work by Björnsson et al. (2015). Both reviews consider quality requirements in terms of positional accuracy as a major challenge in handling large carbon fiber plies due to the tight tolerances on placement, and reference several techniques that attempt to address the problem including 2D Computer Vision (Kühnel et al. 2014), Laser Line Scanning (Gerngross and Nieberl 2016), and Laser Interferometry Based Robot Tracking (Krebs et al. 2016).

While positional accuracy is a challenge, datumed geometric tolerances are well defined. We do not consider positional accuracy of automated composite fabric layup

systems in our study, instead focusing on within-textile inspection tasks based on the FAA recommended criteria such as weave defects, foreign object debris, and interlayer or knit thread defects. Manual fabric inspection by textile experts has been identified as having about 60% accuracy on wide fast moving fabrics (Goyal 2018). This type of inspection is also current practice in industrial carbon fiber textile manufacture, and induces multiple manual inspection points during the CFRP process chain. Automated optical inspection of fabrics (e.g. for garment production) has been studied from the beginning of computer vision, and is widely applied in industrial textile manufacture.

There are two unique challenges in optical dry carbon fiber textile inspection:

- Carbon fibers exhibit highly specular reflection and extremely high absorption of light, leading to dark images with poor contrast if gathered with conventional optical sensor systems
- Reliable detection of defects is a difficult image processing problem due to the wide variety of defect types, variable visual appearance, and extremely limited numbers of defect examples

For segmenting fibers under difficult conditions, approaches such as polarized light filters (Jana 2018) and photometric stereo have been used. We consider the photometric stereo solution proposed by Zambal et al. (2015) as our baseline sensor solution and briefly review it in Sect. 2. Thereafter we focus our efforts on data processing methods on this data to address challenge 2. Detection of defects in textiles has been approached in the literature in a number of different ways, including statistical, spectral, model based, and structural methods. A comprehensive survey on the above mentioned textile fabric defect detection techniques in the literature can be found in the work of Kumar (2008).

Many industrial textile inspection systems with engineered algorithms based on combinations of statistical, structural, and model based methods have been proven to reliably detect general weaving defects such as mispicks (Kumar 2008), and address more specific carbon fiber defects such as fiber orientation (Shi and Wu 2007; Zambal et al. 2015). A key challenge in dry CFRP manufacture is the wide variety of defect classes that must be inspected for, ranging from conventional weaving defects, to foreign objects, to more subtle interlayer defects. This is compounded by the rarity of defects - because of the well-optimized baseline process, defect data is limited and expensive to capture, making engineering of bespoke defect detectors for each defect class difficult, expensive, and time consuming. It is here that learning methods offer a potential solution. As they are data driven they do not require feature engineering for each defect class, saving time and effort. However, the demonstrated strong performance of CNNs applied to supervised image classification (Krizhevsky et al. 2017) is not obviously applicable here as these methods are data intensive and as a basic requirement require many examples of both good and defective data to achieve acceptable real-world accuracy.

In the case of carbon fiber fabric defects, in some cases certain types of defects that have to be inspected for occur so rarely that we may only have one or two naturally occurring examples over a 6 month project observation period. In addition, defect classes like Foreign Object Debris (FOD) may exhibit very large intra-class appearance variation (e.g. metal

shavings and stray carbon fiber strands both constitute FOD), so collecting real data that captures the underlying natural variation may be impossible. Hence, datasets in this domain are highly imbalanced and supervised learning approaches are therefore limited in usefulness. As one cannot hope catalogue the visual appearance variation of a class like FOD, an alternative is to not try to detect FOD specifically at all, but instead simply determine whether a given sample is novel or anomalous when compared to a set of ‘normal’ samples.

The literature on anomaly detection methods concerns techniques to recognize inputs that are significantly different from ‘normal’, where anomalies are rare and not well-defined (Pang et al. 2021). This is an appealing framework to apply to the carbon fiber textile domain, as such methods learn on only ‘normal’ data which is in abundance without requiring any abnormal examples. This method of operation also allows detecting anomalies in novel classes that have never been seen before, potentially providing a general solution to the problem.

There are several approaches to anomaly detection in general, however most techniques in the literature are concerned largely with high intra-class variance - examples that are out of distribution and very different to the normal class. In contrast, in carbon fabric defect detection we are concerned with examples that are generally very visually similar to normal samples except with subtle deviations in small confined regions. It is only recently that industrial defect detection has been considered as an explicit sub problem (Bergmann et al. 2019).

In order to be practical, a carbon fiber textile inspection method must meet the following requirements:

- Require only ‘normal’ data to train
- Achieve extremely high accuracy on real defects
- Allow visual interpretation of results for experts to gain confidence
- Operate fast enough to not bottleneck the cutting or pre-form assembly process

We aim to address these limitations with this study by proposing a real-time novel unsupervised defect detection method for carbon fiber textiles operating on data produced by a photometric stereo sensor system.

The main contributions of our work are:

- We introduce a novel unsupervised defect detection method for dry carbon textiles based on visual transformers and normalizing flows
- We show that when modelling image likelihoods with a normalizing flow, using a visual transformer as a feature encoder outperforms a CNN based feature encoding while eliminating the need for multi-scale and highly augmented inputs due to global attention

- We show that our method achieves state of the art accuracy on our carbon fiber fabric dataset, correctly identifying small stray carbon fibers making up less than 1% of image pixels ( $4 \text{ mm}^2$ ) as anomalous where previous methods fail
- We show that our method is applicable to other industrial domains by reaching state-of-the-art on the publicly available MVTEC-AD industrial texture defect dataset
- We propose a novel method for generating visual anomaly maps using transformer attention that allow interpretation of defect detection results
- We show the method can operate at 30 frames per second

The remainder of this paper is organized as follows. In Sect. 2, we describe the defects under study and the form of the data. In Sect. 3 we review relevant literature on anomaly detection techniques and describe the limitations with respect to carbon fiber textiles. In Sect. 4 we describe our anomaly detection basis and novel additions. The results and discussion are presented in Sect. 5. Finally, Sect. 6 concludes the paper and discusses future work.

## Carbon fiber textile defects and form of the data

### Carbon fiber textile description and defect classes

In aerospace production, textile materials are usually checked through a visual inspection by the material supplier and then again in a second visual inspection by the aerospace manufacturer using the provided material (Schneider 2011). It is this visual inspection we seek to automate in this study. Given the stable process and rarity of incoming defects, layup operators (manual or automated) do not assess a quantitative measure of fabric quality directly, they notice anomalies outside of the expected inherent variation of the fabric, and are trained using a set of visual reference standards to identify defects. If a defect is found, they will discard the material or call for an inspector to disposition rework. For a detector acting upstream of a qualified human inspector who can determine the best course of action based on a specific defect (e.g. type, area, proximity to other defects, location), it is therefore sufficient to identify the presence of an anomaly at the image level instead of the pixel level, and no classification of anomaly type is required. In a high-rate manufacturing context, this use case is emphasized further, as automatically discarding a potentially defective piece of material entirely may be preferable to stopping the line to wait for a manual disposition. This image-level anomaly detection setting is the focus of our study.

We perform our case study on a T700  $2 \times 2$  Twill carbon fiber textile from Hexcel (tm), with a bonded interlayer as



**Fig. 3** The normal condition of the carbon fiber textile under study

**Table 1** Defect Classes

a	Broken or Cut Tow Away from Ply Edge
b	Float or Mispick
c	Frayed Ply Edges
d	Fuzz Ball
e	Hung, Loose, or Pulled in Fill Tow, Kink
f	Interlayer, Missing
g	Interlayer, Melted
h	Missing Tow at Ply Edge
i	Smash
j	Stray Fiber
k	Tow Distortion
l	Visible Foreign Bodies
m	Weave Separation
n	Wrinkle or Crease

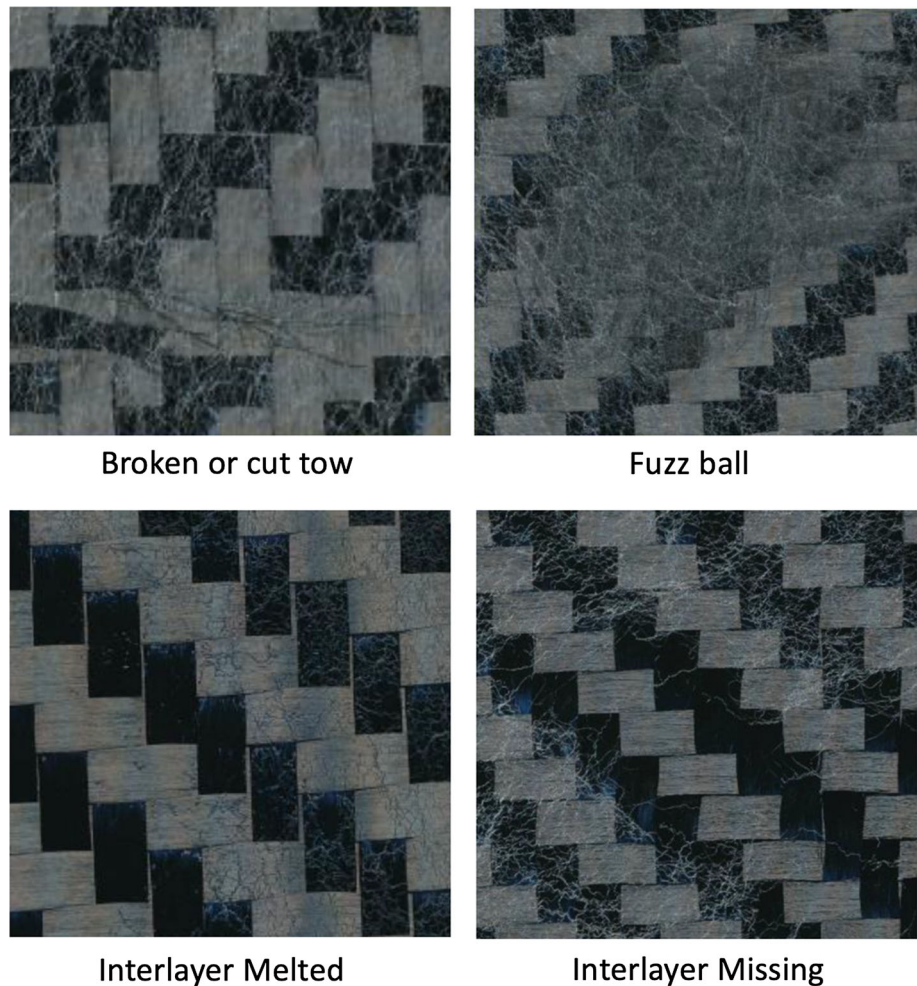
displayed in Fig. 3. The roll width is 1.5 m. The set of fabric defect classes are taken from the FAA recommended criteria for carbon fabrics (Ward et al. 2007), with the addition of two interlayer defect types, (Interlayer Missing and Interlayer Melted).

The defect classes are listed in Table 1.

A selection of defect examples is displayed in Fig. 4.

While some defects are relatively easy for untrained humans to identify such as weave pattern errors, some defects are subtle enough to require significant expertise like interlayer melting. Due to the highly specular and dark carbon fibers and the very faint ‘spiderweb’ interlayer pattern on top, when humans identify these defects, it is common to look at the fabric from multiple angles under light to get contrast. Therefore to achieve reliable automatic anomaly detection, we adopt a data capture solution that has been shown to resolve a number of carbon defects industrially using photometric stereo.

**Fig. 4** Selected Defect Examples



### Photometric stereo

We collect data using a Photometric Stereo system built by Profactor GmbH described by Zambal et al. (2015).

The system uses eight raw images captured from the same point of view with differing illumination patterns in combination with a fiber reflection model based on an infinitesimal cylindrical perfect mirror. The system provides four independent image features for each pixel based on the fiber model (azimuth angle, polar angle, diffuse component, specular component). An example of the raw images and the outputted images on a  $2 \times 2$  twill fabric under study is displayed in Fig. 5. The azimuthal angle image provides a high contrast between fibers running perpendicular to each other and helps resolve the weave pattern. The diffuse image resolves the interlayer without the weave pattern background, and the specular image can be very diagnostic with materials that do not fit either the carbon or interlayer, such as foreign object debris. For details on the photometric stereo algorithms employed we refer readers to the original work by Zambal et al. (2015). We note here that by observation,

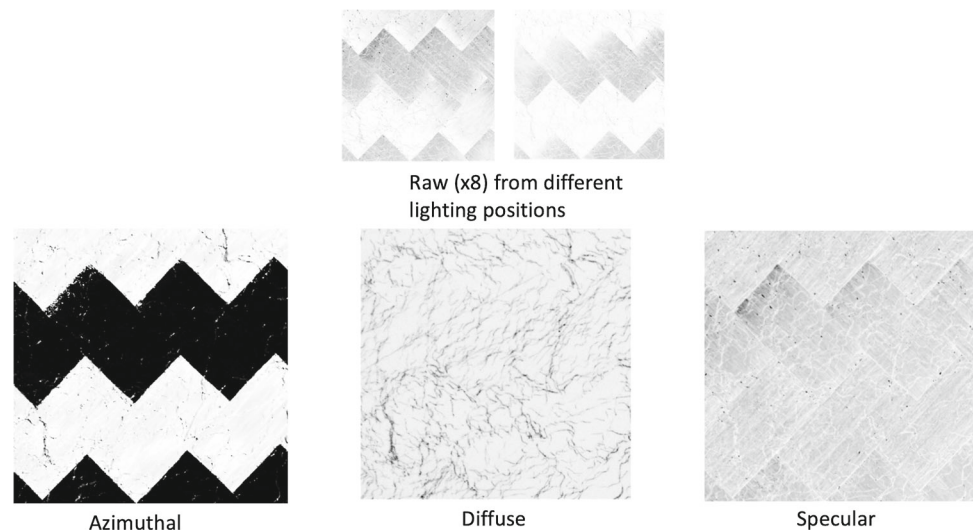
the specular image implicitly contains information about the weave pattern edges, the interlayer, and foreign objects.

Industrial applications of this type of system typically perform defect detection based on image statistics, blob detection or histogram based segmentation of image features such as fiber angle and diffuse brightness. While these can be highly discriminative, they require engineering for each defect class. This is time consuming and requires exhaustive testing due to each defect class being independent. Based on the reported industrial feature engineering success on these feature components (fiber angle, diffuse, and specular) and our observation that the specular image contains relevant information across each feature component, we use this output without feature engineering as the input to our anomaly detector.

### Data format

The carbon fabric data is saved in HDF5 format, with 8 raw images, and 4 feature images for each scan area. We perform limited postprocessing on the data as follows. Stacking

Fig. 5 Raw and Feature Images



azimuthal, diffuse, and specular feature channels on top of each other as an input tensor will cause a problem when using feature extractors pretrained on natural images, as they expect a 3-channel RGB image as input, and the colour channels in our false image will not spatially correlate. As previously noted, the specular image alone to the human eye contains information present in all three types of feature images. We therefore use just the specular image converted to a 3-channel RGB tensor as the input to avoid unnecessary complexity. An example of the raw tensor fed into the anomaly detector that makes this clear is displayed in Fig. 6. The edges of the weave pattern are clearly visible, and the interlayer web appears as black pixels.

Our normal dataset contains 150 scans, each covering a  $50 \text{ mm} \times 50 \text{ mm}$  patch of dry carbon textile, sampled from 2 material batches. Our abnormal dataset used for hold-out val-

idation contains 34 defect images also covering a  $50 \text{ mm} \times 50 \text{ mm}$  area. The abnormal set consists of naturally occurring defects from an aerospace production line during surveillance combined with existing visual defect standard pieces. It covers all defect classes and represents fabric from 5 material batches. This is representative of a realistic data environment where each defect type is outnumbered by normal data by at least 2 orders of magnitude (in this case, 1 or 2 defect samples per class compared to 150 normal samples). The mean defect size in the abnormal sample set was  $100 \text{ mm}^2$ , with a minimum defect size of  $4 \text{ mm}^2$  and maximum defect size of  $256 \text{ mm}^2$ . The mean pixel area covered by defects was 4%. Due to the small size (1% of pixels) and therefore additional difficulty of *stray fiber* defects, we consider two carbon fiber textile datasets - one containing all defect classes, and one with *stray fiber* removed.

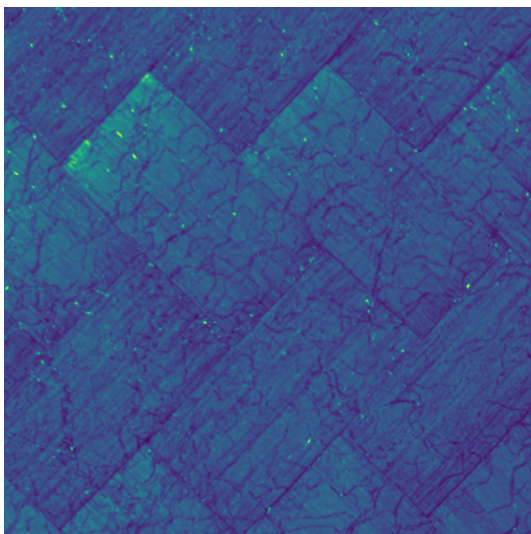


Fig. 6 Stacked and Normalized Input Tensor

## Defect detection

To meet the requirements established in Sect. 1, we do not consider approaches in the literature that use defective examples for training (supervised methods), restricting ourselves to the unsupervised defect detection task, where defects and normal data have similar visual appearance. As interpretability is critical for industrial systems, methods that do not allow visual anomaly maps to be produced are included in our review but not considered as the performance baseline. Due to the lack of literature considering unsupervised anomaly detection for carbon fiber textiles, we evaluate performance using a standard industrial texture defect dataset MVTec-AD (Bergmann et al. 2019), as a proxy for performance on carbon fiber textiles. The MVTec-AD dataset contains defects with similarly small (1% of pixels) defects in the Carpet, and Grid classes.

## Anomaly detection

In general, Anomaly Detection models when given a ‘normal’ dataset and a previously unseen sample, produce an anomaly score representing anomalousness which is then thresholded to obtain the desired false positive rate. We can broadly summarize the literature on unsupervised anomaly detection techniques into two approaches - generative models, and the use of pretrained features.

### Generative models

Anomaly detection using generative models such as Generative Adversarial Networks (GANs) (Akçay et al. 2018) and Convolutional Autoencoders (Bergmann et al. 2018) is based on the idea that such models when trained on only ‘good’ data, will be unable to generate anomalies from the distribution encoded in the latent space. A typical approach is reconstruction, where the output of a generative model is compared to an input query image and pixel-wise differences are used as an anomaly score. In the case of an autoencoder this is trivial, in the case of a GAN, the input query image is run through an inverse generator to get the latent code. Pixels that are very different between the generated image and the original image represent regions of the image that the generative model could not reconstruct, indicating anomalousness. Generative models have been criticized for being sensitive to anomaly size and domain structure. In particular, high frequency normal structures or large anomalies may disproportionately influence the anomaly score (Rudolph et al. 2020; Bergmann et al. 2019), limiting the usefulness of such models in industrial defect detection. The studies by Bergmann et al. (2019, 2021) find that generative anomaly detection models such as GANomaly (Akçay et al. 2018) and f-AnoGAN (Schlegl et al. 2019) perform poorly on industrial textile defects, being unable to reconstruct the required detail when compared to methods based on pretrained features. We therefore focus our efforts on methods using pretrained features for the task at hand.

### Use of pretrained features

There are many approaches making use of the feature space of a pretrained CNN in different ways. Simple techniques such as 1-NN distance in feature space from normal examples as anomaly score (Napoletano et al. 2018), training a 1-class SVM on extracted features from the normal set (Andrews et al. 2016), or fitting probability distributions to extracted features of the normal set (Christiansen et al. 2016). Pretrained features from the final convolution layers of VGG (Simonyan and Zisserman 2014) and AlexNet (Krizhevsky et al. 2017) have been criticised for their robustness on industrial defects (Rippel et al. 2021), however by statistically

combining features from all levels of EfficientNet (Tan and Le 2019), the authors in Rippel et al. (2021) are able to achieve state-of-the-art on some classes, and show that it is theoretically difficult to learn good features from scratch using only normal data, suggesting leveraging pretrained features wherever possible. However, the method proposed by Rippel et al. (2021) is not amenable to visual interpretation, makes significant assumptions around the distribution of extracted image features that are not guaranteed and therefore relies heavily on the particular CNN architecture and training parameters chosen. The authors of DifferNet, Rudolph et al. (2020), propose a model to learn the distribution of features extracted from a pretrained CNN on normal data and produce an image likelihood. This work is reviewed in the following section.

### Normalizing flows

Normalizing flows are a chain of parameterized invertible mappings that can be applied to transform a data distribution into a well defined probability distribution. The authors of Real-NVP, Dinh et al. (2016), proposed a set of invertible and learnable neural network layers to estimate the probability density of images and allow exact log-likelihood computation. The authors of DifferNet, Rudolph et al. (2020), proposed to use this density to detect anomalies, using the Real-NVP normalizing flow layers to transform the distribution of a pretrained feature extractor (AlexNet (Krizhevsky et al. 2017)) to a multivariate normal distribution, using the image likelihood as an anomaly score. DifferNet uses Real-NVP layers with learnable clamping as introduced by Ardizzone et al. (2018). Training is performed by learning to map all ‘normal image’ outputs to a multivariate gaussian with zero mean and identity covariance (ensuring each feature is independent), while penalizing trivial solutions. The anomaly score is then simply the z-score for a given input image. This approach gains the benefits of pretrained features, while enforcing probability distributions over features instead of making a Gaussian assumption as in the work by Rippel et al. (2021) which may not be well founded for all feature extractors. To achieve sensitivity to the scale and appearance of anomalies, the authors perform a multi-scale and multi-orientation feature extraction before the normalizing flow with 64 feature maps at both train and test time. The authors of Rudolph et al. (2020) report no significant difference in performance of their method when replacing the feature extractor with a more powerful one such as ResNet (He et al. 2016). The overall performance of this method is slightly lower than demonstrated in Rippel et al. (2021), however due to the bijectivity it allows anomaly localization, a key requirement of our setting. A technical description of Normalizing Flows and the RealNVP model can be found in Appendix A.

## Performance measure

In non-destructive testing (NDT), a concept called the probability of detection (POD) arises and is estimated as in Eq. 1 (Raymond et al. 2012):

$$POD(a) = \frac{M_d}{N} \quad (1)$$

where  $a$  is the defect size,  $M_d$  is the number of defects of size  $a$  detected, and  $N$  is the total number of defects of size  $a$ . The POD measure is used on a defect class basis by introducing artificial flaws of a certain type in several sizes. An estimation of POD requires a full enumeration of defect types, and a large distribution of naturally occurring defects, or creating artificial defects that could be drawn from the natural defect distribution. For fabric defects this is highly non-trivial, as the visual appearance of defects is wide and the detection probability is a function of much more than size, making representative data difficult to obtain.

While avoiding the cost and effort of performing a full POD study with artificial defects is a primary motivator in this work, and not all defect classes have defined area limits (some are simply disallowed upon human visual inspection), we report POD values at  $6.5 \text{ mm} \times 6.5 \text{ mm}$  as a commonly used aerospace defect threshold (Roach and Rice 2014), and  $2 \text{ mm} \times 2 \text{ mm}$  as a visual identification threshold for unallowable defects.

To allow direct comparison to other methods in the literature and compare performance on a publicly available dataset, we also report a standard measure of classification accuracy as follows. In order to arrive at a final judgement

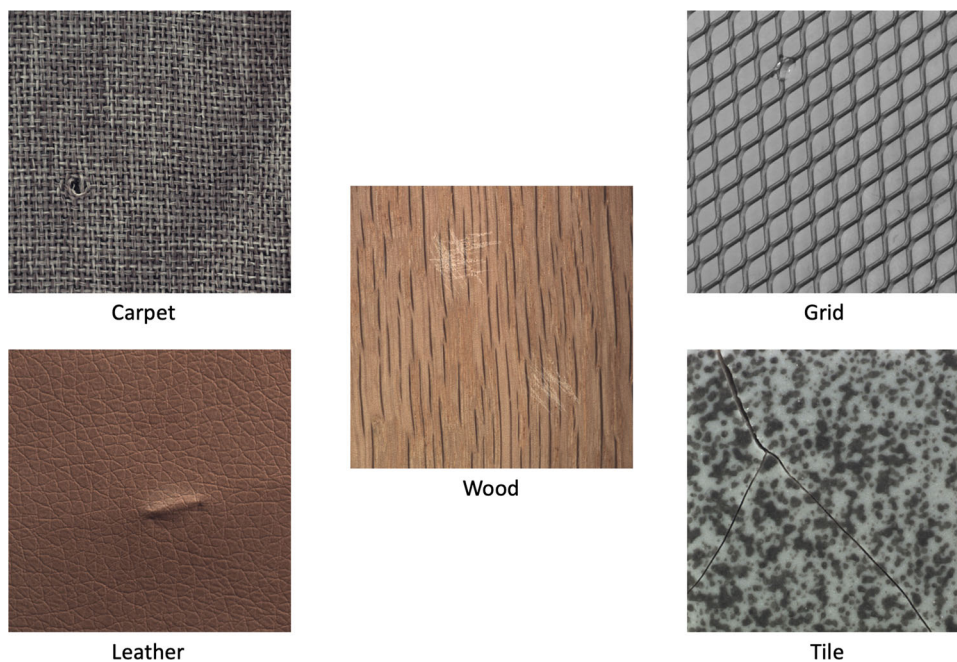
on the anomalousness of a given input, a threshold is applied to the anomaly score produced by the model. The threshold trades off the True Positive Rate (TPR), which is the ratio of the cases properly classified as abnormal to the real abnormal cases, with the False Positive Rate (FPR), which is the ratio of the cases falsely classified as abnormal to the real normal cases. The ideal TPR and FPR would be 1 and 0; however this tradeoff of sensitivity and specificity is a feature of every anomaly detector. To evaluate the general performance, we therefore construct the receiver operating characteristics (ROC) curve, that plots the TPR against the FPR by changing the threshold of the classifier. The area under the receiver operating characteristics (AUROC) can be used to interpret how the detector distinguishes the boundary between the anomalousness and normality. An AUROC of 1.0 means that all abnormalities were classified as abnormal, and no normal samples were classified as anomalous.

## Limitations of existing models

The MVTec-AD (Bergmann et al. 2019) Texture dataset provides an excellent baseline for industrial material inspection tasks, and a good proxy for performance on carbon fabric. The dataset contains five texture classes, Grid, Leather, Tile, Carpet, and Wood (See Fig. 7).

While achieving good results on average, DifferNet (Rudolph et al. 2020) performs poorly on Grid and Carpet relative to other classes. Grid and Carpet exhibit sometimes subtle and small defects on top of a large region of normality, which to the human eye provides a clear backdrop on which to assess anomalousness. When testing on our carbon fabric

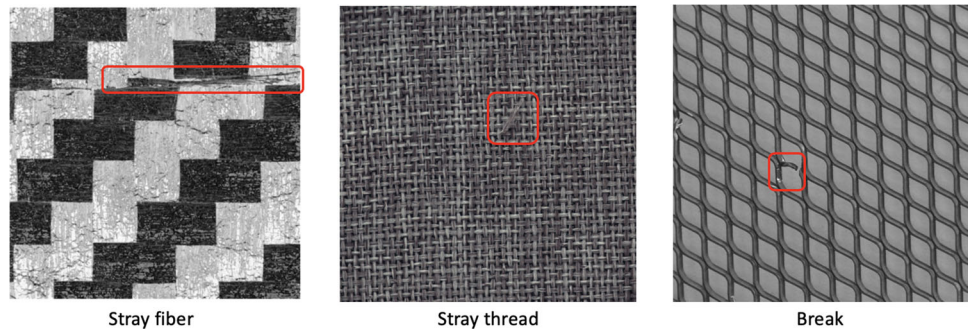
**Fig. 7** MVTec-AD Texture Defect Examples





**Table 2** DifferNet performance (%AUROC) on small subtle defects on regular backgrounds

Dataset	DifferNet (Rudolph et al. 2020)
MVTec-AD Texture (Average)	94.6
MVTec-AD Carpet	92.9
MVTec-AD Grid	84.0
Carbon Fabric Without Stray Fibers (Minimum Defect Size 25 mm <sup>2</sup> )	100
Carbon Fabric (Minimum Defect Size 4 mm <sup>2</sup> )	97.5

**Fig. 8** DifferNet failure modes for small subtle anomalies

defect dataset, we find that DifferNet performs well in all defect classes except for those including stray fibers, where similarly to Grid and Carpet, small and subtle defects are present on a large ‘normal’ background (See Table 2 and Fig. 8). CNN features have been criticized for the receptive field size for image level anomaly detection (Rippel et al. 2021). While the model uses multi-scale multi-orientation feature extraction to improve sensitivity to defect scale and aggregate context, this is clearly insufficient for these datasets. We hypothesize that the lack of global context for the CNN features is the primary cause of this limitation.

## Methodology

### Visual transformers

Transformers (Vaswani et al. 2017), first proposed for machine translation, achieve state of the art results across a variety of natural language processing (NLP) tasks. The key difference between transformers and other architectures such as Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs), is that while CNNs have a receptive field and RNNs have a context window, limiting what input information can be used to calculate a particular output, transformers can simultaneously gather information from their entire input - so called global attention. This mechanism is very general, and it has been shown that a multi-head self-attention layer with sufficient number of heads can be at least as expressive as any convolutional layer (Cordonnier et al. 2019). Recently introduced Visual transformers (Dosovitskiy et al. 2020) (ViT) apply the transformer archi-

ture on images by breaking up an input image into a grid of small image patches, each  $16 \times 16$ . These models achieve state of the art classification results, beating CNNs like ResNet (He et al. 2016) when trained on an extremely large dataset (Google’s JFT-300M (Hinton et al. 2015)). As the self-attention layers in each transformer encoder allow the network to use global context to inform its output for every image patch, the authors show that the Visual Transformer can learn relevant context to ignore irrelevant parts of images and therefore outperform CNNs. Applications of Visual Transformers outside of classification are so far uncommon, although wide application of Visual Transformers and similar models where CNNs are currently used is likely. A technical description of Transformer models in general and Visual Transformers is available in Appendix A.

### Proposed model

We propose that using a Visual Transformer encoder instead of a CNN to extract image features for texture anomaly detection will improve performance due to global context. In Rippel et al. (2021), the authors comment that different types of anomalies show different variabilities across feature scales, and that the pretrained CNN layers may not surface relevant information at the last layer. We hypothesize that compared to the CNN used in DifferNet, the global attention mechanism of the ViT model should eliminate the need for multi-scale and highly augmented inputs. In particular, for texture anomalies where the image itself is made of a ‘normal’ background with an ‘abnormal’ region, we hypothesize that based on the classification attention maps produced by Dosovitskiy et al. (2020), the pretrained self-attention mech-

anism will naturally attend to anomalous regions as they are more object-like than the background, making the transformer features more powerful than CNN features for texture anomalies and improving performance.

While the ViT architecture was designed for classification, we propose to use the output of a pretrained visual transformer encoder as input to a normalizing flow for image likelihood calculation and then anomaly detection.

The proposed model architecture is displayed in Fig. 10. It is based on the same core idea as DifferNet. That is, we estimate the likelihood of image features  $y \in Y$  from the anomaly-free training images  $x \in X$ . This is performed using the same RealNVP-based normalized flow as in DifferNet, mapping all normal samples  $x$  to a multivariate gaussian distribution.

We use the ViT-Base  $16 \times 16$  model as described by Dosovitskiy et al. (2020) with 12 transformer layers pretrained on JFT-300M with the MLP head removed, taking a  $384 \times 384$  image as input and producing a 768 dimensional feature vector.

A diagram of the normalizing flow blocks used is displayed in Fig. 9. We use 8 coupling blocks with 2048 neurons each and a clamping parameter of  $\alpha = 3$ . The weights of the ViT encoder are fixed during training.

The NF is trained to map all non-anomalous samples  $x$  as close to  $z = 0$  as possible using the negative log likelihood of a standard normal distribution as the loss function  $\mathcal{L}(x)$  (see Eq. 2).

$$\mathcal{L}(x) = \frac{\|z\|_2^2}{2} - \log \left| \det \frac{\partial z}{\partial x} \right| \tag{2}$$

In Eq. 2,  $z$  is the z-score of  $x$  from a standard normal distribution, and  $x$  is a given non-anomalous sample.

Unlike Rudolph et al. (2020) our model does not require optimizing over multiple transformations of an input image, nor do we need to aggregate features extracted at multiple scales or use data augmentation, resulting in a significantly simpler pipeline.

For more details on the normalizing flow construction we refer readers to the work by Rudolph et al. (2020) and Ardizzone et al. (2018).

While the invertibility of the normalizing flow allows us to retrieve the image gradient with respect to the output as a kind of interpretable anomaly map, the complicated self-attention mechanism of transformers means that we cannot treat the feature extractor as a fixed set of filters and so cannot expect good anomaly localization. We therefore propose to use gradients through the normalizing flow combined with self-attention rollout maps aggregated across all layers of the transformer encoder to provide interpretable anomaly maps for each image. This allows a human to visually introspect the parts of an image that were most influential in determining its anomalousness. While not a complete explanation of the output, it is easily interpretable by a human and valuable in production use. The details are presented in Sect. 5.1.2.

### Anomaly localization

#### Attention rollout

Abnar and Zuidema (2020) show that the raw attention weights from transformer layers are not useful to explain transformer outputs because the self-attention mechanism combines information from one layer to the next, mixing inputs in complicated ways. To compute the relevance of inputs with respect to the output of the transformer, they propose a simple method to model the information flow through a transformer called Attention Rollout. Attention Rollout considers the weight of each edge between nodes as the proportion of information flowing between those nodes, and then multiplies the edge weights along all possible paths in the network to determine how much the information at a given input node is propagated to a given output node.

To compute the attentions from  $l_i$  to  $l_j$ , Attention Rollout recursively multiplies the attention weights matrices in all the layers below as in Eq. 3.

$$\tilde{A}(l_i) = \begin{cases} A(l_i)\tilde{A}(l_{i-1}) & \text{if } i > j \\ A(l_i) & \text{if } i = j \end{cases} \tag{3}$$

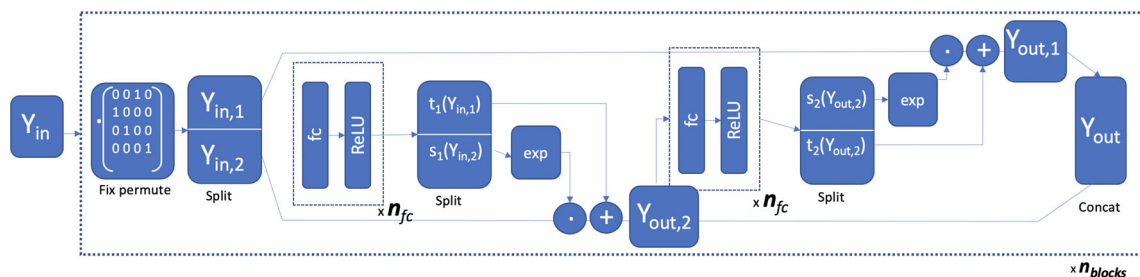


Fig. 9 Normalizing Flow Block from DifferNet (Rudolph et al. 2020)

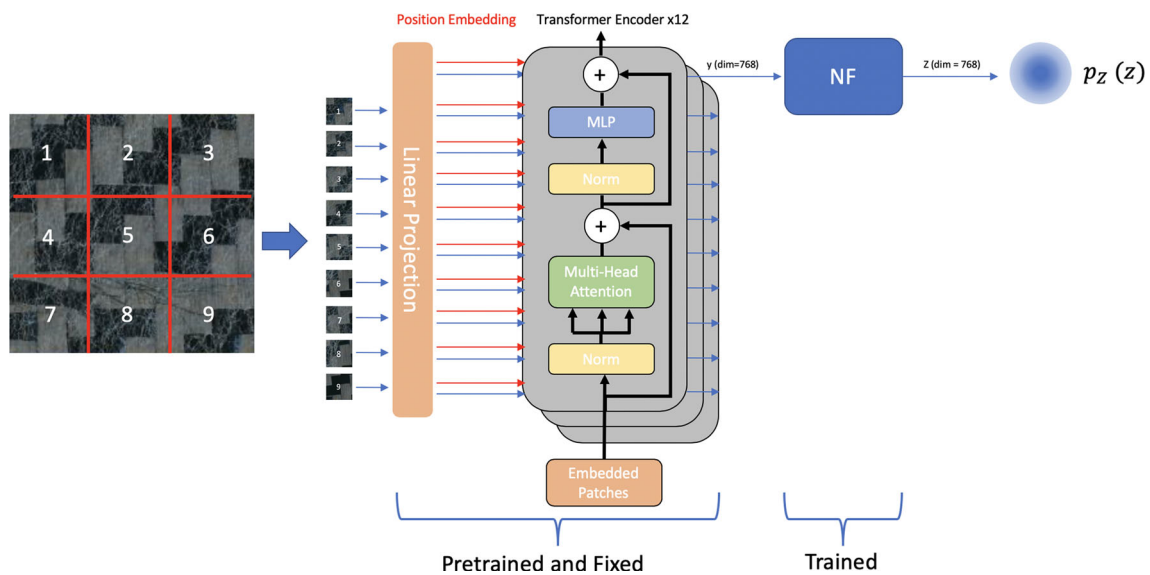


Fig. 10 Proposed Anomaly Detection Architecture

where  $\tilde{A}$  is the Attention Rollout Matrix,  $A$  is the Raw Attention Weights Matrix and  $j = 0$  is set to compute attention from the input layer.

### Training setup

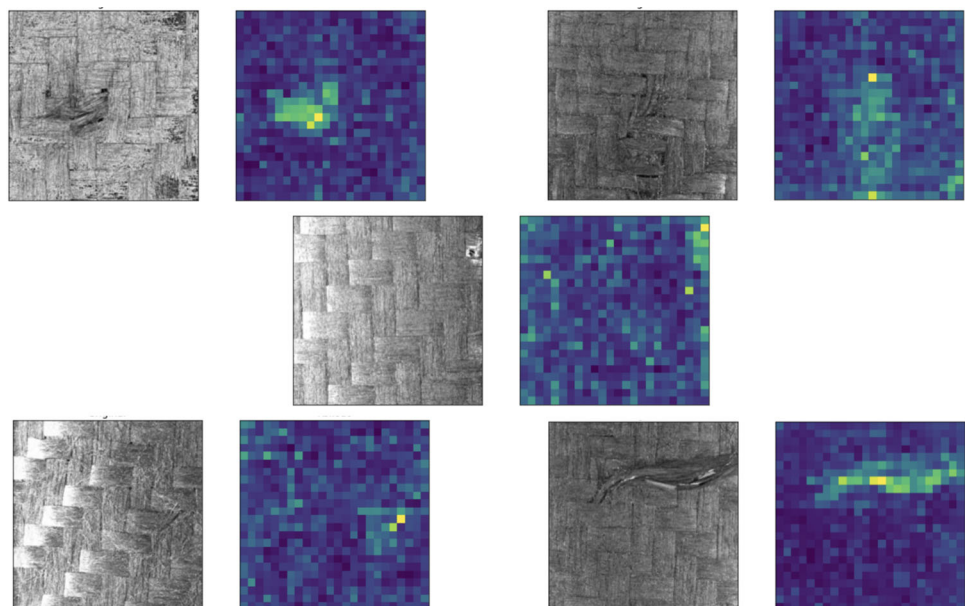
We train the model for 48 epochs with a batch size of 8 on an NVIDIA V100 GPU. The model is built in PyTorch (Paszke et al. 2019), based on open source implementations of DifferNet (Rudolph et al. 2020) and ViT (Wightman 2019). Learning is done using the Adam Optimizer (Kingma and Ba 2014) with a learning rate of  $2e-5$ . The final model tested is

the model with the best validation accuracy across all epochs. The training takes approximately 20 minutes.

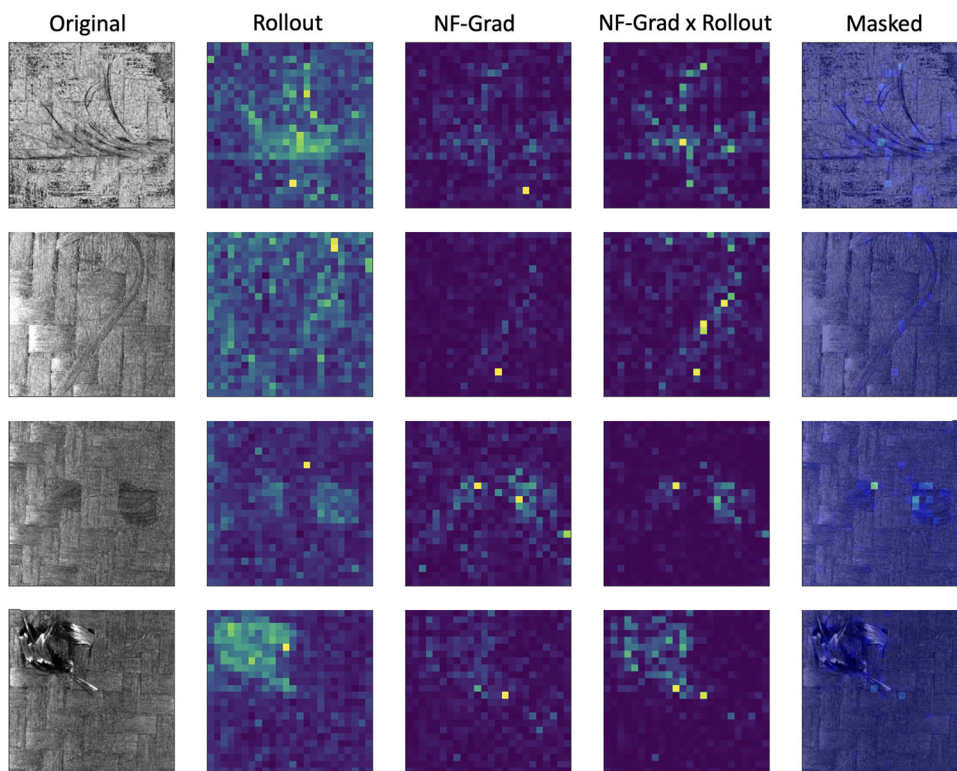
### Results and discussion

We report the performance of our anomaly detector on our carbon fiber textile defect dataset and, separately, a publicly available industrial texture dataset from MVTec-AD to allow comparison with other approaches. The same pretrained ViT encoder is used for all datasets, and the normalizing flow is trained from scratch on each dataset, and then evaluated. As the training occurs upfront only once on ‘normal’ data, and

Fig. 11 Zero-Shot attention on anomalous regions



**Fig. 12** Anomaly maps produced through self-attention rollout and gradient weighting



industrial deployment of the model in a given data domain does not require retraining, the relevant performance metric is inference time. The inference time of our model 32 ms per image, allowing 30 FPS realtime deployment on a modern GPU.

### Carbon fiber fabric performance

#### Zero shot attention performance on carbon fabric

We hypothesized that the pretrained transformer would use global context to infer texture anomalies by attending to ‘interesting’ object-like parts of the image, rather than the background. Using Attention Rollout, we find that this is indeed true for texture anomalies in our dataset (See Fig. 11). However, Attention rollout is performed without any regard to the classification of the input image and so by itself is not an interpretable view of an anomaly detector.

#### Anomaly maps on carbon fabric

Attention rollout allows us to look at what parts of the image most contributed to the information flow from the image to the outputted features. Inverting the normalizing flow and the feature extractor to retrieve the raw image gradient with respect to the output as in DifferNet is less useful with a transformer than a CNN as the locality of features is violated, however the gradient is still useful to weight regions

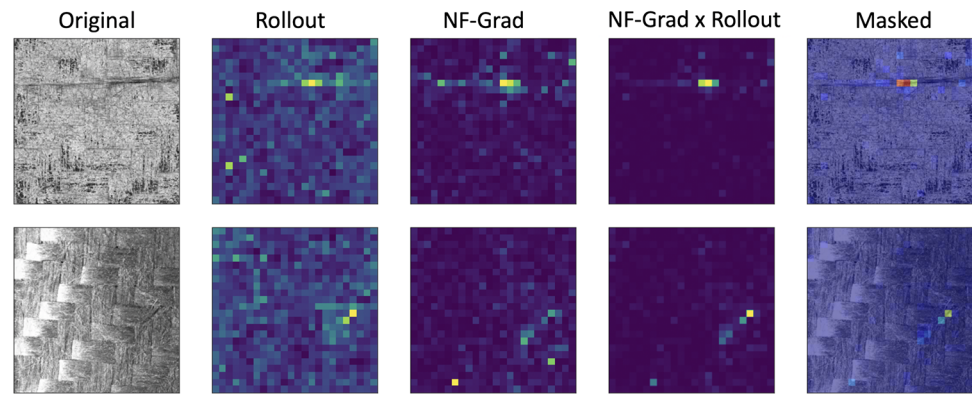
**Table 3** Performance on Carbon Fiber Fabric (%AUROC)

Dataset	DifferNet (Rudolph et al. 2020)	Ours
Carbon Fabric (Minimum Defect Size 4 mm <sup>2</sup> )	97.5	100
Without Stray Fibers (Minimum Defect Size 25 mm <sup>2</sup> )	100	100

of global attention, as anomalies are usually local. We therefore propose to create an anomaly map by multiplying the gradient through the normalizing flow with the attention rollout of the ViT. We find it produces a more convincing and interpretable anomaly map than either alone (See Fig. 12).

We achieve perfect performance on our carbon fabric dataset, solving the task for these defect classes in this setting (See Table 3). Our model is able to perfectly distinguish anomalies without any false positive rate tradeoff even on challenging stray fiber defects. We hypothesized that the ViT model would improve performance due to global attention allowing greater context on texture anomalies, especially when anomalies are small and subtle. We indeed report better performance and our anomaly maps indicate our sensitivity to small defects. Examples of subtle stray fiber anomalies scored as such by our model are presented in Fig. 13.

**Fig. 13** Stray fibers clearly scored by our model that are ambiguously scored by DifferNet



**Table 4** Performance on Carbon Fiber Fabric (POD)

Model	POD 43 mm <sup>2</sup>	POD 4 mm <sup>2</sup>
DifferNet (Rudolph et al. 2020)	1.0	0.0
Ours	1.0	1.0

## Industrial textures

Our method reaches state of the art on all MVTEC-AD texture classes except Wood where we have comparable performance. Overall, classes excluding grid perform above 99% AUROC. Of note, our method improves detection on difficult classes like Grid and Carpet by 7-10%, and 4% on average, validating that the performance improvement from our proposed method is observed on other industrial texture anomaly detection problems. In Table 5 we reproduce the MVTEC-AD results of the anomaly detection techniques presented by Bergmann et al. (2019) and Rudolph et al. (2020) to allow direct comparison against our results. We replicated the experiments for DifferNet shown in Rudolph et al. (2020) as the previous state-of-the-art, the other results are reproduced directly from the tables in the work of Bergmann et al. (2019).

## Limitations of the method and future research

Our method is designed for the imbalanced data setting where very few examples of defects are available. It has been shown

to perform well in terms of AUROC and POD on industrial data, however our POD estimations are based on a relatively small sample size. Future work could collect naturally occurring defects off a high-rate production line over a long period of time to collect further diversity in defect sizes and improve the POD estimate while not introducing artificially sized unrealistic data to the distribution.

## Conclusion

In this paper, we present an unsupervised defect detection method for carbon fiber textiles that meets four key criteria for industrial applicability: data needs, accuracy, interpretability, and speed. Building on previous work, we for the first time combine Visual Transformers and Normalizing Flows to gather global context from input images and directly produce an image likelihood score. As defect samples are expensive and difficult to collect, our method requires only normal examples to train. We demonstrate that our method correctly detects 100% of anomalies with a 0% false positive rate on a industrial carbon fabric dataset with 34 real defect samples, including subtle stray fiber defects where previous methods have been shown to fail. We validate these results on the public defect dataset *MVTEC-AD Textures*, where we also achieve state of the art results, outperforming previous work by 4% overall and 10% on challenging classes with small defects, proving the applicability of our method to other

**Table 5** Comparative Performance on MVTEC-AD Texture Defects Dataset (%AUROC)

Class	GANomaly Akçay et al. (2018)	OCSVM Andrews et al. (2016)	1-NN Napoletano et al. (2018)	DifferNet Rudolph et al. (2020)	Ours
Grid	70.8	41.0	55.7	84.0	92.2
Leather	84.2	88.0	90.3	97.1	100
Tile	79.4	87.6	96.9	99.4	99.6
Carpet	69.9	62.7	81.1	92.9	99.3
Wood	83.4	95.3	93.4	99.8	99.6
Average	76.2	71.9	83.9	94.6	98.1

domains. Additionally, we propose a method to extract interpretable anomaly maps from Visual Transformer Attention Rollout and Image Likelihood Gradients that produces convincing explanations for detected anomalies, enhancing trust in the model for industrial use. Finally, we show that the inference time for the model is acceptable at 32 ms, allowing 30 frames per second evaluation and achieving real-time performance.

**Acknowledgements** The authors would like to thank Dr. Luke Fletcher for review.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions. This research was funded by Australian Government Commonwealth Supported Postgraduate Place under the Research Training Scheme. Support from Boeing was limited to data provision and accommodation of Study Leave.

**Data availability** The carbon fiber dataset was collected subject to a confidentiality agreement and is not publicly available, please contact the authors. The MVTec-AD dataset can be retrieved under license from <https://www.mvtec.com/company/research/datasets/mvtec-ad>.

**Code availability** The code will be made available on publication at <https://www.github.com/mszarski/cfrp-anomaly-detection-jim-code>

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A

### A.1 Transformers

#### A.1.1 Attention

Learning dependencies between distant input regions is facilitated by the concept of ‘attention’, as way to focus on certain elements in the input and output while retaining information about the surrounding context (Luong et al. 2015). Attention in neural networks is typically a vector of learned weights that encode the correlation or interdependence between either the input and output elements or within the input elements (called ‘self-attention’) (Bahdanau et al. 2015). When mak-

ing a prediction, the attention vector is used to weight the contributions of the elements.

#### A.1.2 Multi-head self attention

The Transformer (Vaswani et al. 2017) architecture uses a mechanism called scaled dot-product attention, based on an information retrieval concept using a *query* to index into a set of *keys* each of which is assigned a *value*. The dot product of the Query Matrix  $Q$  and Key Matrix  $K$  is taken to assign a weight to the Value Matrix  $V$ . Keys that are most similar to the query will be proportionally weighted higher, and then the output is simply the weighted sum of the values (See Eq. 4). The Transformer uses self-attention, which means the queries, keys, and values all come from the same input (usually embedded in some way before being turned into the input  $Q$ ,  $K$ , and  $V$  matrices). (4):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (4)$$

The Transformer gathers information across its global inputs with an approach called Multi-head Self Attention (MHSA), using  $h$  self-attention ‘heads’ evaluated in parallel. This is done for each head with a separate linear projection so each head can focus on different information at different positions.

#### A.1.3 The transformer architecture

The Transformer architecture combines MHSA, Layer Normalization (LN) (Ba et al. 2016), feed-forward networks (FFNs), and skip connections (He et al. 2016) into two types of Transformer Blocks: The Encoder, and the Decoder. Copies of these blocks are then stacked to create the final encoder and decoder modules. For sequence-to-sequence learning like machine translation for which the Transformer was originally designed, an encoder-decoder framework is used, where the encoder reads the input and outputs a representation which is then fed to the decoder which autoregressively generates the output sequence. For tasks where the output does not need to be an autoregressively generated sequence, just the encoder output can be used.

The overall transformer architecture is displayed in Fig. 14.

#### A.1.4 Vision transformers

Recently introduced Vision Transformers (Dosovitskiy et al. 2020) (ViT) apply the transformer architecture on images by breaking up an input image into a grid of small image patches, each  $16 \times 16$ , and encoding those pixels into a patch embedding. The global self-attention in the ViT architecture

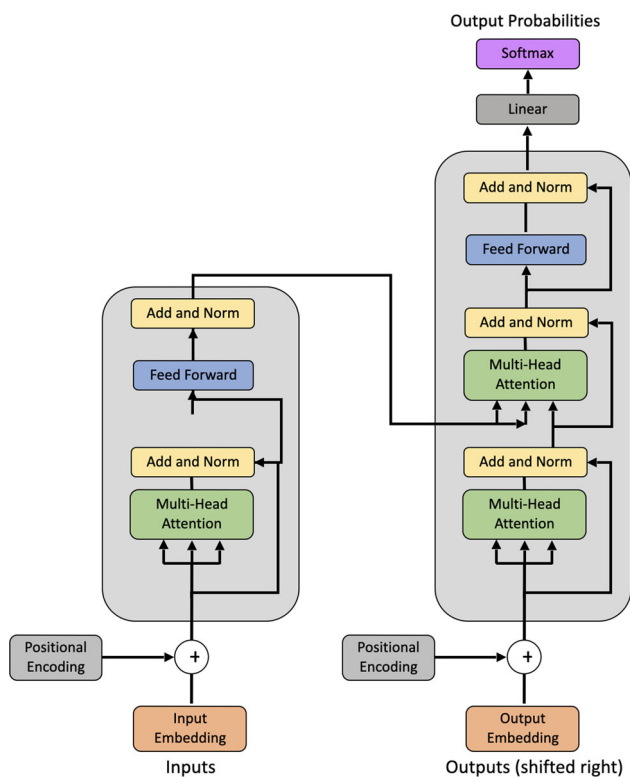


Fig. 14 The Transformer Architecture (Vaswani et al. 2017)

means that it does not suffer from limited receptive fields like CNN models, as it can gather information from all locations in the image at once to make its prediction.

### A.1.5 The vision transformer architecture

The Vision Transformer was proposed as a classification model, and as such it does not need or use an encoder-decoder framework. Instead it uses only the encoder portion of a transformer to produce an image encoding that is then classified by an MLP head. Because Transformers are designed to process sequences of inputs, the image is re-shaped into a sequence of flattened 2D patches before being linearly embedded and then fed into a standard transformer encoder. A class token is prepended to the sequence of image patches as in BERT (Devlin et al. 2018), to add an additional readout that is independent of the other patch embeddings. The class token output will contain all the information required for the downstream classification head, taking into account all areas of the image. The class token output (only) is then fed through an MLP to make the final prediction. As opposed to the original Transformer model, the positional encoding is learned, not fixed.

The overall Visual Transformer (ViT) architecture is displayed in Fig. 15.

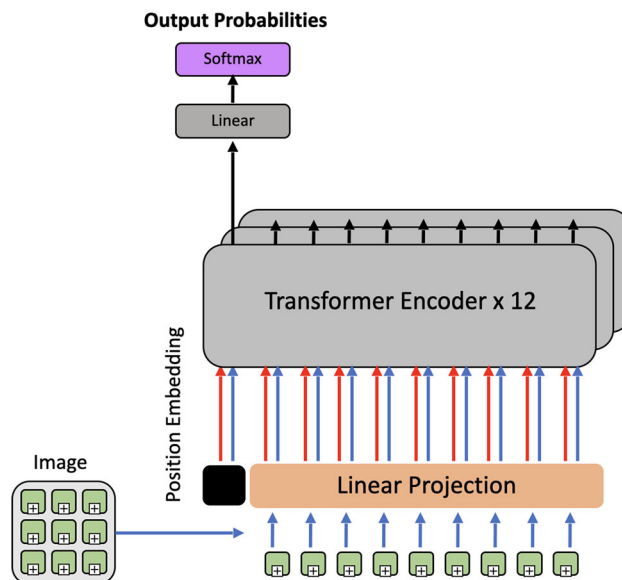


Fig. 15 The Visual Transformer Architecture (Dosovitskiy et al. 2020)

### A.2 Normalizing flows

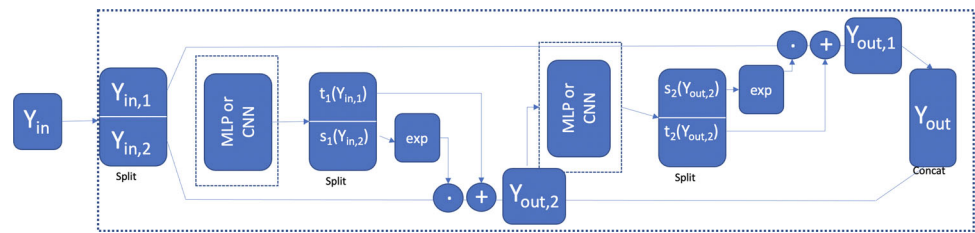
Normalizing Flows, introduced by Tabak and Turner (2013), and first used with neural networks by Rezende and Mohamed (2015), map a simple distribution (say, a standard Gaussian for easy sampling and density evaluation) to an arbitrary complex distribution (learned from data). This is achieved by applying a sequence of invertible transformation functions  $f_i$  in succession to the simple distribution. In practice, the latent distribution  $p_Z(\mathbf{z})$  is commonly chosen to be a standard Gaussian. As the probability density function of a transformed variable can be computed using the change of variables formula (See Eq. 5), by applying this repeatedly between each pair of successive transformations we form a chain and eventually obtain a probability distribution of the complex target variable. This chain  $f = f^K \circ f^{K-1} \circ \dots \circ f^1$  is called a *Normalizing Flow*. Given observed data  $\mathbf{x} \in X$ , a  $p_Z$  on a latent variable  $\mathbf{z} \in Z$ , and a bijection  $f : X \rightarrow Z$ , the change of variable formula defines a distribution on  $X$ :

$$p_X(x) = p_Z(f(x)) \left| \det \left( \frac{\partial f(x)}{\partial x^T} \right) \right| \tag{5}$$

$$\log(p_X(x)) = \log(p_Z(f(x))) + \log \left( \left| \det \left( \frac{\partial f(x)}{\partial x^T} \right) \right| \right) \tag{6}$$

where  $\frac{\partial f(x)}{\partial x^T}$  is the Jacobian of  $f$  at  $x$ .

**Fig. 16** A RealNVP coupling layer



Therefore, with initial distribution  $\mathbf{z}_0$  and normalizing flow  $f$  we have:

$$\mathbf{x} = \mathbf{z}_K = f^K \circ f^{K-1} \circ \dots \circ f^1(\mathbf{z}_0) \tag{7}$$

$$\begin{aligned} \log(p(\mathbf{x})) &= \log(p_{Z_K}(\mathbf{z}_K)) \\ &= \log(p_{Z_0}(\mathbf{z}_0)) - \sum_{i=1}^K \log \left| \det \frac{\partial f_i}{\partial \mathbf{z}_{i-1}} \right| \end{aligned} \tag{8}$$

To train the normalizing flow, we then directly maximize the log-likelihood (See Eq. 8) of the training data  $x$  with respect to the parameters  $\theta$  that parameterize the series of invertible transformations  $f$ . We can then sample from the resulting distribution simply by drawing a  $z$  and computing its inverse image  $x = f^{-1}(z)$ . Similarly, To compute the likelihood of a given  $x$ , we simply compute  $f(x)$  and multiply it by the determinant of the Jacobian.

For for this to be possible, two conditions must hold on  $f_i$ :

- $f_i$  must be invertible
- Computing the determinant of the Jacobian of  $f_i$  needs to be easy.

**A.2.1 RealNVP**

The RealNVP (Dinh et al. 2016) (Real-valued Non-Volume Preserving) model used Normalizing Flows to successfully model natural images through sampling and log-likelihood evaluation. To achieve this, RealNVP builds a Normalizing Flow out of a sequence of bijective *affine coupling layers*  $f_{\text{aff}}$ . The input dimension is first split into two parts, with the first  $d \in D$  dimensions left untouched, and the second  $d + 1$  to  $D$  dimensions undergo a **scale**  $s(\cdot)$ , and **translation**  $t(\cdot)$  affine transformation, where the scale and translation parameters are taken from the first  $d$  dimensions of the input  $x$ .

The transformation performed by each layer for input  $x$  and output  $y$  is given in Eq. 9.

$$\begin{aligned} f_{\text{aff}}^{-1}(x_{1:d}, x_{d+1:D}) &= (y_{1:d}, y_{d+1:D}), \\ \begin{cases} y_{1:d} &= x_{1:d} \\ y_{d+1:D} &= (x_{d+1:D} + t(x_{1:d})) \odot \exp(s(x_{1:d})) \end{cases} \end{aligned} \tag{9}$$

where the scale and shift functions  $s(\cdot)$  and  $t(\cdot)$  are implemented by a neural network. In the case of RealNVP for image modelling this is a CNN, however this can be any feed-forward neural network and MLPs are used to demonstrate RealNVP on other types of data.

A visual representation of a single coupling layer is shown in Fig. 16.

Passing the input through several of these layers in an alternating pattern, such that the components that are left unchanged in one coupling layer are updated in the next allows a complex, nonlinear mapping to be learned.

These affine coupling layers are easily inverted because the part of the input vector that determines the scaling and translation is preserved, and computing the inverse does not require computing the inverse of  $s$  or  $t$ , so these functions can be arbitrarily complex.

Likewise, because computing the Jacobian determinant does not involve computing the Jacobian of  $s(\cdot)$  or  $t(\cdot)$  the required Jacobian computation is also tractable:

$$\begin{aligned} \log \left| \det \frac{\partial f_{\text{aff}}^{-1}}{\partial x} \right| &= \sum_{i=1}^{\dim(x_{d+1:D})} s(x_{1:d})_i. \end{aligned} \tag{10}$$

In some Normalizing Flow models using affine coupling layers, the split of the input is based on a fixed permutation, and not on an index, but all of the above still applies to this case. For a more detailed introduction to normalizing flows based on coupling layers, please see the work by Kobyzev et al. (2020).

**References**

Abnar, S., & Zuidema, W. (2020). Quantifying attention flow in transformers. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 4190–4197

Akçay, S., Atapour-Abarghouei, A., & Breckon, T. P. (2018). Ganomaly: Semi-supervised anomaly detection via adversarial training. *Asian conference on computer vision* (pp. 622–637). Berlin: Springer.

Andrews, J., Tanay, T., Morton, E. J., & Griffin, L. D. (2016). Transfer representation-learning for anomaly detection. *JMLR*



- Angerer, A., Ehinger, C., Hoffmann, A., Reif, W., Reinhart, G., & Strasser, G. (2010). Automated cutting and handling of carbon fiber fabrics in aerospace industries. In: 2010 IEEE international conference on automation science and engineering, IEEE (pp. 861–866)
- Ardiszone, L., Kruse, J., Rother, C., & Köthe, U. (2018). Analyzing inverse problems with invertible neural networks. In: International conference on learning representations
- Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450)
- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In: Bengio, Y., & LeCun, Y., (eds) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, [arxiv:1409.0473](https://arxiv.org/abs/1409.0473)
- Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., & Steger, C. (2018). Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint [arXiv:1807.02011](https://arxiv.org/abs/1807.02011)
- Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9592–9600).
- Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., & Steger, C. (2021). The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4), 1038–1059.
- Björnsson, A., Jonsson, M., & Johansen, K. (2015). Automation of composite manufacturing using off-the-shelf solutions, three cases from the aerospace industry. In: ICCM20-The 20th international conference on composite materials, 19–24 July 2015, Copenhagen, Denmark
- Christiansen, P., Nielsen, L. N., Steen, K. A., Jørgensen, R. N., & Karstoft, H. (2016). Deepanomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, 16(11), 1904.
- Cordonnier, J. B., Loukas, A., & Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. In: International conference on learning representations.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
- Dinh, L., Sohl-Dickstein, J., Bengio, S. (2016). Density estimation using real nvp. arXiv preprint [arXiv:1605.08803](https://arxiv.org/abs/1605.08803)
- Dosovitskiy, A., Beyler, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929)
- Elkington, M., Ward, C., & Sarkytbayev, A. (2017). Automated composite draping: A review. In: SAMPE, SAMPE North America
- Gerngross, T., & Nieberl, D. (2016). Automated manufacturing of large, three-dimensional cfrp parts from dry textiles. *CEAS Aeronautical Journal*, 7(2), 241–257.
- Goyal, A. (2018). 4 - automation in fabric inspection. In: Nayak R, Padhye R (eds) Automation in Garment Manufacturing, The Textile Institute Book Series, Woodhead Publishing (pp. 75–107). <https://doi.org/10.1016/B978-0-08-101211-6.00004-5>, <http://www.sciencedirect.com/science/article/pii/B9780081012116000045>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).
- Heuer, H., Schulze, M., Pooch, M., Gäbler, S., Nocke, A., Bardl, G., Cherif, C., Klein, M., Kupke, R., Vetter, R., et al. (2015). Review on quality assurance along the cfrp value chain-non-destructive testing of fabrics, preforms and cfrp by hf radio wave techniques. *Composites Part B: Engineering*, 77, 494–501.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531)
- Jana, P. (2018). 9 - automation in sewing technology. In: Nayak, R., & Padhye, R. (eds) Automation in Garment Manufacturing, The Textile Institute Book Series, Woodhead Publishing (pp. 199 – 236). <https://doi.org/10.1016/B978-0-08-101211-6.00009-4>, <http://www.sciencedirect.com/science/article/pii/B9780081012116000094>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Kobyzev, I., Prince, S., & Brubaker, M. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*
- Krebs, F., Larsen, L., Braun, G., & Dudenhausen, W. (2016). Design of a multifunctional cell for aerospace cfrp production. *The International Journal of Advanced Manufacturing Technology*, 85(1), 17–24.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Kühnel, M., Schuster, A., Buchheim, A., Gerngroß, T., & Kupke, M. (2014). Automated near-net-shape preforming of carbon fiber reinforced thermoplastics (cfrtp). In: ICS conference journal.
- Kumar, A. (2008). Computer-vision-based fabric defect detection: A survey. *IEEE Transactions on Industrial Electronics*, 55(1), 348–363.
- Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. arXiv preprint [arXiv:1508.04025](https://arxiv.org/abs/1508.04025)
- Napoleitano, P., Piccoli, F., & Schettini, R. (2018). Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18(1), 209.
- Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2), 1–38.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems 32, Curran Associates, Inc. (pp. 8024–8035). <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- Raymond, B. M., et al. (2012). Numerical analysis of probability of detecting defects in engineering materials. In: 18th world conference on nondestructive testing
- Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In: International conference on machine learning, PMLR (pp. 1530–1538).
- Rippel, O., Mertens, P., & Merhof, D. (2021). Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th international conference on pattern recognition (ICPR), IEEE (pp. 6726–6733).
- Roach, D. P., & Rice, T. M. (2014). *A quantitative assessment of advanced ndi techniques for detecting flaws in composite laminate aircraft structures*.
- Rudolph, M., Wandt, B., & Rosenhahn, B. (2020). Same same but different: Semi-supervised defect detection with normalizing flows. arXiv preprint [arXiv:2008.12577](https://arxiv.org/abs/2008.12577)
- Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., & Schmidt-Erfurth, U. (2019). f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54, 30–44.
- Schneider, M. (2011). 5 - automated analysis of defects in non-crimp fabrics for composites. In: Lomov, S. V. (ed) Non-Crimp Fabric Composites, Woodhead Publishing Series in Composites Science

- and Engineering, Woodhead Publishing (pp. 103–114). <https://doi.org/10.1533/9780857092533.1.103>, <https://www.sciencedirect.com/science/article/pii/B9781845697624500059>
- Shi, L., & Wu, S. (2007). Automatic fiber orientation detection for sewed carbon fibers. *Tsinghua Science and Technology*, 12(4), 447–452.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
- Tabak, E. G., & Turner, C. V. (2013). A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2), 145–164.
- Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint [arXiv:1905.11946](https://arxiv.org/abs/1905.11946)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In: *Advances in neural information processing systems* (pp. 5998–6008).
- Ward, S., McCarvill, W., & Tomblin, J. (2007). *Guidelines and recommended criteria for the development of a material specification for carbon fiber/epoxy fabric prepregs*. Federal Aviation Administration: Office of Aviation Research and Development.
- Wightman, R. (2019). Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, <https://doi.org/10.5281/zenodo.4414861>
- Zambal, S., Palfinger, W., Stöger, M., & Eitzinger, C. (2015). Accurate fibre orientation measurement for carbon fibre surfaces. *Pattern Recognition*, 48(11), 3324–3332.

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