Controlled Synthesis of Fibre-reinforced Plastics Images from Segmentation Maps using Generative Adversarial Neural Networks

Nicolas Schaaf[1][a], Hans Aoyang Zhou[2][b], Chrismarie Enslin[2][c], Florian Brillowski[3][d] and Daniel Lütticke[2][e]

[1] *RWTH Aachen University, Aachen, Germany*

[2] *Information Management in Mechanical Engineering, RWTH Aachen University, Aachen, Germany*

[3] *Institute of Textile Technology, RWTH Aachen University, Aachen, Germany*

Keywords: GAN, Deep Learning, Fibre-reinforced Plastics, Quality Assurance, Defect Detection, Data Augmentation.

Abstract: The replacement of traditional construction materials with lightweight fibre-reinforced plastics is an accepted way to reduce emissions. By automating quality assurance, errors in production can be detected earlier, avoiding follow-up costs and hard-to-recycle scrap. Deep learning based defect detection systems have shown promising results, but their prediction accuracy often suffers from scarce labelled data in production processes. Especially in the domain of fibre-reinforced plastics, the task remains challenging because of varying textile specific errors. In our work, we applied conditional generative adversarial networks combined with image-to-image translation methods to address data scarcity through generating synthetic images. By training a generative model on image-segmentation pairs, we produce realistic fibre images matching the given segmentation maps. Our model enables control over generated output images of arbitrary fibre shapes and structures, including gaps, ondulations, and folds as error classes. We evaluate our synthetic images based on GAN metrics, feature distribution and show that they are suitable as a data augmentation method to improve the error classification performance of deep neural networks. Thereby, we provide a solution for the manufacturing domain of fibre-reinforced plastics with scarce data, consequently contributing to an automated defect detection system that reduces resource-intensive scrap in the future.

1 INTRODUCTION

Climate change, legal emission regulations and increasing environmental awareness in society lead to a steadily growing demand for lightweight materials (Türe and Türe, 2020). The underlying idea of using these materials is to minimise the mass of moving parts, thereby reducing energy consumption, as well as emissions. Fibre-reinforced plastics (FRP) are a material class that provides the necessary requirements of mechanical strength and at the same time low weight (Miao and Xin, 2018). However, producing FRP is very expensive due to their resource-intensive manufacturing process. Therefore, FRP manufacturers need to avoid scrap and have extensive quality control (QC) in order to be profitable and competitive.

Currently, available QC systems can only indirectly detect defects, such as folds (Fuhr, 2017) by measuring the fibre orientations with for example Canny-Edge detection. If a deviation in fibre orientation is detected, the specific defect is then determined by a time-consuming, manual visual inspection. One possible solution to reduce manual labour is to use Deep Learning (DL) models for automated inspection to directly localise and classify defects in semifinished textile products.

The technical feasibility of DL approaches for QC in manufacturing of FRP has already been proven for various approaches, such as using Convolutional Neural Networks (CNNs) for semantic segmentation in order to detect defects in fabric image data (Wei et al., 2019; Jing et al., 2019; Sacco et al., 2018; Sacco et al., 2019). All approaches have a data-intensive labelling process in common, in order to achieve high detection accuracy.

[a] https://orcid.org/0000-0002-7844-3229
[b] https://orcid.org/0000-0002-7768-4303
[c] https://orcid.org/0000-0002-3047-5846
[d] https://orcid.org/0000-0003-4805-9573
[e] https://orcid.org/0000-0002-7070-8018

Schaaf, N., Zhou, H., Enslin, C., Brillowski, F. and Lütticke, D.
Controlled Synthesis of Fibre-reinforced Plastics Images from Segmentation Maps using Generative Adversarial Neural Networks.
DOI: 10.5220/0010913700003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 801-809
ISBN: 978-989-758-547-0; ISSN: 2184-433X

Despite the promising results from the literature, the usage of DL models as an automated solution for image-based QC is still not standardised in industrial environments, particularly not for the manufacturing industry of FRP. On one hand, many FRP manufacturing companies have machine parks consisting of old devices that lack data-gathering sensors or interfaces. On the other hand, due to small production lot sizes, available data is also limited. As a result, there is no, or only incomplete data available for the different steps of the production process. The accessible data is often insufficient for training of detection models with acceptable detection accuracy. Additionally, manual annotation of image data is an expensive and time-consuming process.

An established solution to address insufficient amounts of data is through data augmentation. Most commonly, it uses existing images to synthetically generate similar but realistic images via, e.g. geometric transformations or stochastic noise. One recent learning based approach for generating new data samples are Generative Adversarial Networks (GAN) (Goodfellow et al., 2014). By learning the underlying data distribution of the available data, they have shown remarkable results in generating realistic images.

Therefore, we investigate in this contribution the effectiveness of GANs as a data augmentation method for training DL models in the data scarce environment of manufacturing FRP. In our work, we do not only show that GANs are well-suited for generating realistic looking images of FRP, but also semantic meaningful images for training a defect classifier without manual labelling. In our evaluation, we validate necessary use case driven modifications of model architecture and training process, qualitatively and quantitatively. Through our quantitative results we show, that the generated images on one hand contribute positively in error detection accuracy and the other hand resemble the feature distribution of the available dataset. Our contribution validates GANs as a data augmentation method for industrial applications with scarce labelled data like the manufacturing of FRP.

2 RELATED WORK

Closely related to our task of synthesising images of FRP, is the task of texture synthesis. The first approach that used a CNN to synthesise textures was introduced in (Gatys et al., 2015). The main idea was to describe textures as the correlation between high-level features acquired through feature maps generated by different layers of a pretrained CNN. Further improvements in generating texture with repetitive patterns were addressed by adding spatial transformation (e.g. flipping or translation) on the feature maps (Berger and Memisevic, 2017) and constraining the frequency domain acquired with Fourier transformation of the generated image (Liu et al., 2016), or a combination of both (S. Schreiber et al., 2016).

The first GAN based texture synthesis was introduced in (Jetchev et al., 2017). In their work, the authors replaced the usual noise vector with a spatial noise tensor as the input for the generator. This allowed the generation of high-quality images that scale to bigger synthesised image sizes. Further improvements in controlling the synthesis of images with GANs were presented in (Xian et al., 2018). The authors developed a conditional GAN that generates images of objects based on a sketch. Furthermore, different parts of the image can be modified through a texture patch.

Outside texture synthesis, GANs are used as a data augmentation method in different domains with scarce labeled data (Shorten and Khoshgoftaar, 2019). This is especially crucial for problems with imbalanced label distributions (e.g. anomaly detection). One popular domain where these types of problems frequently appear is the field of medical imaging (Yi et al., 2019). Here, training samples are limited due to their labeling cost or patient privacy, which motivated researchers to investigate GANs as an alternative for synthesising training samples. Likewise, in (Zhu et al., 2018) the authors successfully use GANs to augment training data of underrepresented label classes for emotion classification. Their results show that through the data augmentation, the classification performance increased significantly. In the domain of textile manufacturing, GANs are investigated to generate textile design patterns (Fayyaz et al., 2020). In their work, the authors evaluate different state-of-the-art GAN-based architectures based on their inception score and use a style transfer approach to generate a combination of multiple textile designs. Although GANs show promising results as a data augmentation method, our work is the first to apply GANs as data augmentation method in the context of manufacturing of fibre-reinforced plastics.

3 METHOD

3.1 Dataset

Our dataset is created from greyscale surface scans of plain weaves made of carbon fibres. Angle and distance of the scanner head varies over the different

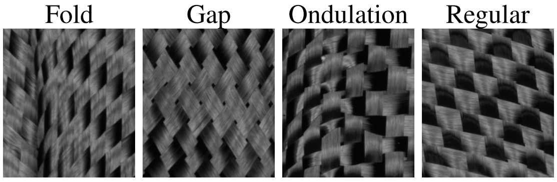

Figure 1: Overview of different error classes of real images.

scans. To form our classification dataset, the surface scans are cropped into 384 × 341 patches, filtered, and assigned manually to one of four classes, namely fold, gap, ondulation and regular (cf. Figure 1). As certain types of errors often occur together with other errors, we define the prioritisation with decreasing order as fold, ondulation, gap, and regular. In the case of multiple errors we assign the class with the highest priority. The prioritisation is based on observations that folds often come along with ondulations and ondulations often come along with gaps. The "regular" class only contains defect-free images. Our full dataset X^R_{full} contains more than 12k instances.

In this work, we use the superscript R for real and G for synthetic (generated) datasets and instances. For the experiments we split up X^R_{full} into 3 parts, X^R_{train}, X^R_{val}, X^R_{test}, for training, validation, and testing of models, respectively. Table 1 provides a detailed overview of the number of instances in all datasets used in our research.

Table 1: Overview of instances per dataset.

Dataset	Fold	Gap	Ondulation	Regular
X^R_{full}	2717	5861	1261	2460
$X^R_{\text{train,min}}$	42	89	19	35
X^R_{train}	906	1954	421	820
X^R_{val}	906	1954	420	820
X^R_{test}	905	1953	420	820
$X^G_{\text{U-Net}}$	5000	5000	5000	5000
X^G_{StarGAN}	5000	5000	5000	5000

3.2 Image-to-Image Approach

Given the goal of using GANs as a data augmentation method for defect classification, the generation of images need to be adaptable according to different error classes. For this purpose, and to capture the geometric structure of fibre parts, we create segmentation maps, i.e. pixel-wise class labels, from our image data. To simplify the manual work, fibre edges are approximated by quadratic or cubic Bézier curves, or linear splines.

The occurring patterns between warp and weft are encoded with two different colours, depicted as orange and blue in Figure 2, each indicating a fibre orientation. Note that, since warp and weft cannot be differentiated in the image patches, the fibre colours are interchangeable. In addition to the fibre colour, we add a third colour indicating fold lines, while gaps are indicated as black sections (cf. Figure 2).

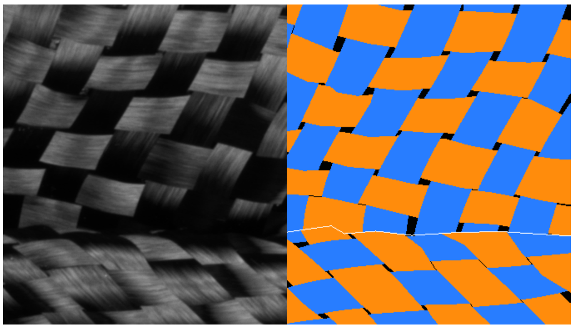

Figure 2: Segmentation map annotation of weave.

With the derived pairs of segmentation maps and real images we train a generator, similarly to an image-to-image approach that translates segmentation maps to realistic images. This training approach with corresponding segmentation maps enables the synthesis of specific weave images and the precise control of fibre structure and geometry. For the training of the generator we use a modified implementation of BicycleGAN (Zhu et al., 2017) to learn a model that maps segmentation maps to weave images.

BicycleGAN is a multimodal, supervised image-to-image approach, which works as a hybrid of GANs and autoencoders. Thus, it trains on input-output pairs of images and allows, through the modification of latent variables, to generate images in different styles. These attributes make it the suitable framework for our use case. In their implementation, Zhu et al. (2017) use an adaptation of U-Net (Ronneberger et al., 2015) as generator. U-Net is an architecture that has been developed and become popular for image segmentation tasks. It consists of an encoder-decoder "bottleneck" architecture that has been extended by skip-connections from contracting to expanding layers. A similar, but unsupervised, image-to-image framework to BicycleGAN is StarGAN v2 (Choi et al., 2020). Like BicycleGAN it is multimodal, but, as a major difference, it works also for unpaired data and can be trained on images from more than two domains. Unlike Zhu et al. (2017), Choi et al. (2020) use an encoder-decoder architecture without skip-connections for their generator.

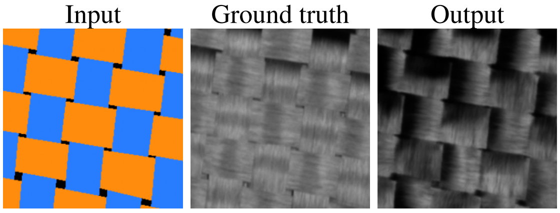

Figure 3: Example of the orientation problem, where fibres of the output image are oriented towards the wrong direction.

4 IMPLEMENTATION DETAILS

When training BicycleGAN on our dataset in the configuration of Zhu et al. (2017), we were facing several issues, such as blurry outputs or the "orientation problem". The orientation problem appears as fibre parts oriented in the wrong direction, i.e. rotated 90 degrees with respect to the direction intended by the segmentation map (cf. Figure 3). We addressed these problems with a slightly modified architecture and training setup.

4.1 Architectures

For the generator, we tested two different architectures: An extended version of the U-Net architecture proposed by Zhu et al. (2017), namely "Extended U-Net", and a modified version of the StarGAN v2 generator architecture proposed by Choi et al. (2020).

4.1.1 Extended U-Net

In our experiments, we found that the U-Net architecture proposed by Zhu et al. (2017) suffers from the orientation problem when trained against an unconditional discriminator. The problem could not be solved by additional training or an increased number of parameters. It appears that an unconditional discriminator is unable to learn the coherence of correct fibre orientation, as it only gets to see the generator outputs. The problem can be addressed with a conditional discriminator, i.e. a discriminator which is fed input-output pairs. Thereby, it is able to learn the correct coherence from real image pairs and provide a corrective feedback to the generator. However, we observed that a conditional discriminator in combination with the U-Net architecture leads to significant lower image quality and blurry results. Our experiments indicate that this is a general problem of U-Net-like architectures.

As an alternative to address the orientation problem while keeping an unconditional discriminator, we found it useful to place residual blocks in front of U-Net. We orient towards the head architecture of ResNet-34 (K. He et al., 2016) and add a 7×7 convolutional layer with 64 filters and a stride of 1, and three residual blocks in front of U-Net to create an "Extended U-Net". The residual blocks correspond to the ResNet-34 residual block with batch normalization and ReLU activation replaced by instance normalisation and LeakyReLU, respectively. We observed that although the orientation problem still occurred in early phases of training, the modification overcomes it with longer training. Unlearning through overtraining is also possible, therefore an optimal stopping point is found.

4.1.2 StarGAN V2

In addition to our proposed "Extended U-Net" architecture, we also experimented with the StarGAN v2 generator architecture to address the orientation problem. We found that for this architecture only a conditional discriminator leads to acceptable results, as for an unconditional discriminator the generator failed to match the given segmentation maps in its outputs.

For our implementation, we slightly modified the architecture by adding a downsampling and upsampling block, respectively, to shrink the bottleneck to a size of $8 \times 8 \times 512$. Additionally, we removed the adaptive wing based heatmaps from the skip connections, as our dataset does not require an alignment of faces, and do not use a latent mapping network, as we only have a single target domain.

4.2 Training

To show that our approach works for small dataset sizes, we train our generator on a small dataset $X^R_{\text{train,min}}$ consisting of 185 images from X^R_{train} and their corresponding segmentation maps. The models are trained on 256×256 patches randomly cropped from the training images. Before cropping, we scale images larger by a factor of 1.1 to 1.33, since it leads to improved image quality. During the test phase, we scale segmentation maps larger by a factor of 1.3, feed them into the generator, and scale the output back to the original size. Training samples are augmented through rotating, flipping, and commuting the fibre channels of the segmentation maps. We represent the colours in our segmentation maps in terms of one-hot-encoding instead of RGB values in order to prevent erroneously implied proximity by numerically close colour values. Due to the increased number of parameters of our generator architectures, we train the Extended U-Net generator for 1200 epochs and the StarGAN v2 generator for 1600 epochs during the normal training phase and fine-tuning phase, respectively. Unlike Zhu et al. (2017), we use the same discriminator for the two training cycles, since the usage of two independent discriminators does not yield any benefits in our experiments.

5 EVALUATION

For the evaluation of our synthesised images we build on four different methods: a manual, qualitative inspection by eye, an evaluation based on GAN scores, an evaluation of the feature distribution, and a use case based evaluation using synthetic images as augmentation in a defect classification task. To create our synthetic images, we generate 5000 synthetic segmentation maps by modelling fibre courses with lines or quadratic Bézier curves and adding class-specific characteristics, i.e. gaps, ondulations, or folds, in various shapes. We translate the same 5000 segmentation maps once with the Extended U-Net and the StarGAN v2 generator to create $X^G_{\text{U-Net}}$ and X^G_{StarGAN}, respectively. For our evaluation we assume that the class intended to imitate in the segmentation map is equivalent to the real class of the corresponding generator output.

5.1 Manual Evaluation

To evaluate qualitative criteria and gain a first impression of the generated image quality, we assess our synthetic images by eye. We found that both generator architectures produce sharp and realistic results, without any evidence of the previously introduced orientation problem. Blending of segmentation maps and output images (cf. Figure 4) reveals that both architectures produce matching results, while the Extended U-Net architecture produces perfectly aligned images and the results of StarGAN v2 deviate up to 4 pixels from segmentation inputs in some cases.

Additionally, we evaluate the influence of the latent vectors by generating several results from each segmentation map with different latent vectors. We found that both generators are able to cover the full spectrum of appearances in real images, i.e. variations in lighting and shadows (cf. Figure 5) and Figure 6.

5.2 Evaluation on GAN Scores

A manual evaluation is not feasible to capture the entire variance of generator outputs, as that would mean inspection of thousands of images. Moreover, the variety in possible outputs makes it difficult to assess which dataset is of higher quality. To address these problems, we use GAN scores, more specifically: Inception Score (IS)(Salimans et al., 2016), Fréchet

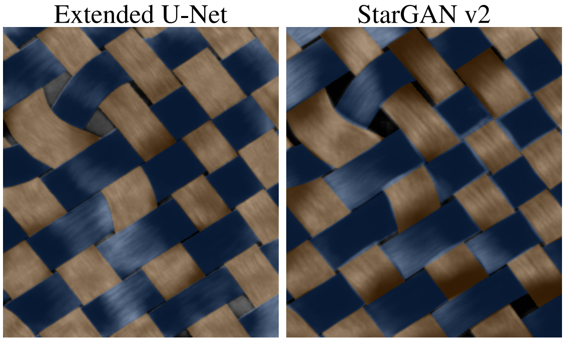

Figure 4: Blending of segmentation maps with output images.

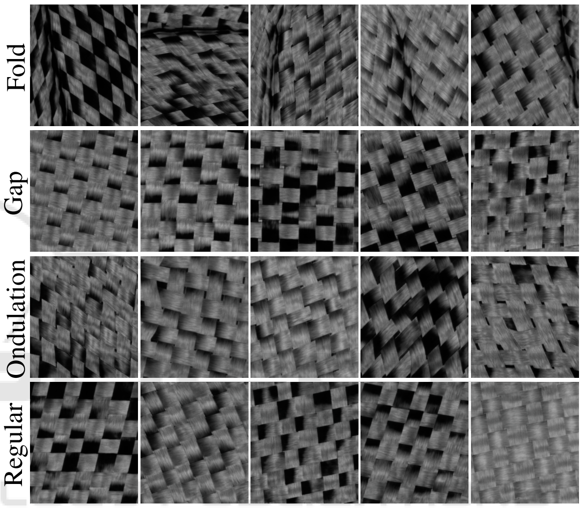

Figure 5: Results of Extended U-Net generator architecture.

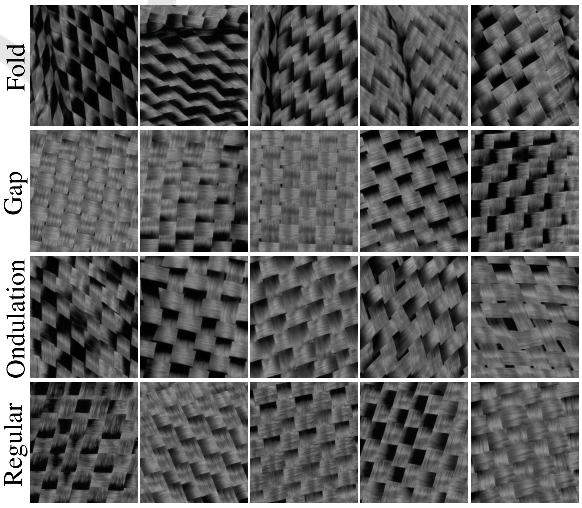

Figure 6: Results of StarGAN v2 generator architecture.

Inception Distance (FID)(Heusel et al., 2017), and their conditional extensions Between-class (BCIS) and Within-class (WCIS) IS, and Between-class (BCFID) and Within-class (WCFID) FID (Benny et al., 2021). The conditional extensions provide additional meaning for our specific approach, as our combination of segmentation map synthesis and image-to-image generator can be interpreted as a class-conditional image generator modelling $p(x|c)$, with the segmentation map synthesis modelling the distribution $p(s|c)$ of segmentation maps s conditioned on conditional classes c and the image-to-image generator modelling $p(x|s)$. Since our real dataset is imbalanced, we slightly modify scores by weighting samples depending on the number of instances per class in order to ensure that each class contributes the same amount to the final score. This ensures that every error class is equally important for the given use case.

5.2.1 Inception Score

To capture if our synthetic images match distinct classes, we use the Inception Score (IS) (Salimans et al., 2016). In their work, the authors defined the score as the expectation of the Kullback-Leibler divergence between the true class posterior $p(t|x)$ and the true class prior $p(t)$ over the distribution $p(x)$ of images x defined by the generator or real dataset:

$$\mathrm{IS} = \exp\left(\mathbb{E}_{x \sim p(x)} D_{\mathrm{KL}}(p(t|x) \parallel p(t))\right) \tag{1}$$

Usually, the expectation is estimated as an average over all instances $x^{(i)}$. To ensure that each class has the same influence on the score result, we replace the average by the following weighted average scheme:

$$\mathrm{IS} \approx \exp\left(\frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i_c=1}^{N_c} D_{\mathrm{KL}}(p(t|x^{(i_c)} \parallel \hat{p}(t))\right)$$
$$\hat{p}(t) = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i_c=1}^{N_c} p(t|x^{(i_c)}). \tag{2}$$

Hence, we first calculate an average over the N_c instances within each class c then an average over the C class averages. Thereby, we avoid domination of the score result by classes with many instances. We calculate the IS on 10 Inception v3 models (Szegedy et al., 2016) trained on X^R_{train} and average the IS results, since the IS is reported to show high variances between different training runs(Barratt and Sharma, 2018). The training procedure is identical to the procedure used in 5.4. In the same manner of the IS, we modify the BCIS and WCIS by assuming that the class priors are equally distributed, i.e. $p(c) = \frac{1}{C}$.

Table 2: Inception scores and FID scores (compared to X^R_{val}) for different datasets.

Dataset	IS	BCIS	WCIS	FID	BCFID	WCFID
X^R_{test}	3.30	2.81	1.28	0.14	0.06	0.42
$X^G_{\text{U-Net}}$	2.22	1.64	1.35	41.38	37.93	51.76
X^G_{StarGAN}	2.20	1.48	1.49	61.00	60.11	74.30

X^R_{test} achieves significant higher scores than both synthetic datasets, while scores of $X^G_{\text{U-Net}}$ and X^G_{StarGAN} only differ on a minor level (cf. Table 2). Note that since we modified the final Inception v3 layer to predict 4 classes, the IS has an upper bound of 4 instead of 1000 (Barratt and Sharma, 2018).

5.2.2 Fréchet Inception Distance

To compare real and synthetic images on a feature distribution level, we use the Fréchet Inception distance (FID)(Heusel et al., 2017). The FID is defined as the Fréchet distance between the (assumed) normal distribution of Inception v3 feature vectors with mean vectors μ^R, μ^G and covariance matrices Σ^R, Σ^G from real and generated data, respectively:

$$\mathrm{FID} = \|\mu^R - \mu^G\|_2^2 + \mathrm{Tr}\left(\Sigma^R + \Sigma^G - 2(\Sigma^R\Sigma^G)^{\frac{1}{2}}\right) \tag{3}$$

Analogous to our modification of the IS, we adapt the estimate of the mean vector $\hat{\mu}$ and the covariance matrix $\hat{\Sigma}$ to

$$\hat{\mu} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i_c=1}^{N_c} f(x^{(i_c)}), \tag{4}$$

and

$$\hat{\Sigma}_{m,n} =$$
$$\frac{N}{(N-1)C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i_c=1}^{N_c}(f(x^{(i_c)})_m - \hat{\mu}_m)(f(x^{(i_c)})_n - \hat{\mu}_n), \tag{5}$$

with $f(x)$ denoting the extracted feature vector of an instance x and N the total number of instances in the dataset. In the same manner, we modify the BCFID and WCFID by assuming that the class priors are equally distributed, i.e. $p(c) = \frac{1}{C}$. To obtain feature vectors relevant for the use case, we extract them from an Inception v3 models trained on X^R_{train} instead of the original model trained on ImageNet(Szegedy et al., 2016; Deng et al., 2009) and average the score results from 10 different training runs. As a reference dataset representing the real data we use X^R_{val}. The FID scores (cf. Table 2) indicate that both synthetic datasets differ significantly from the real dataset in terms of features, while the feature distribution of $X^G_{\text{U-Net}}$ is more similar than the one of X^G_{StarGAN}.

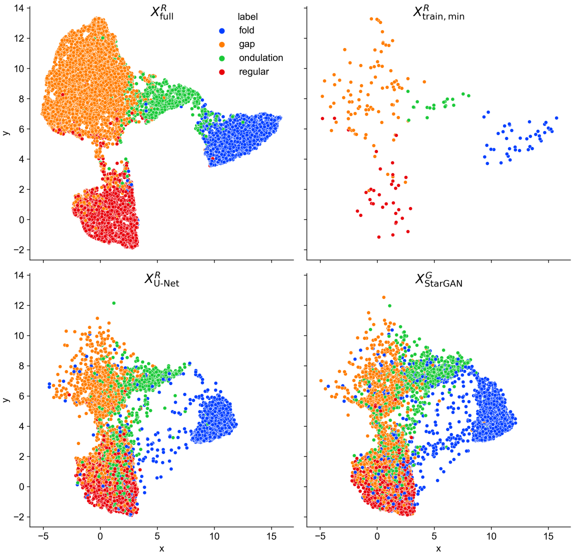

Figure 7: Feature distribution mapped in 2D using UMAP.

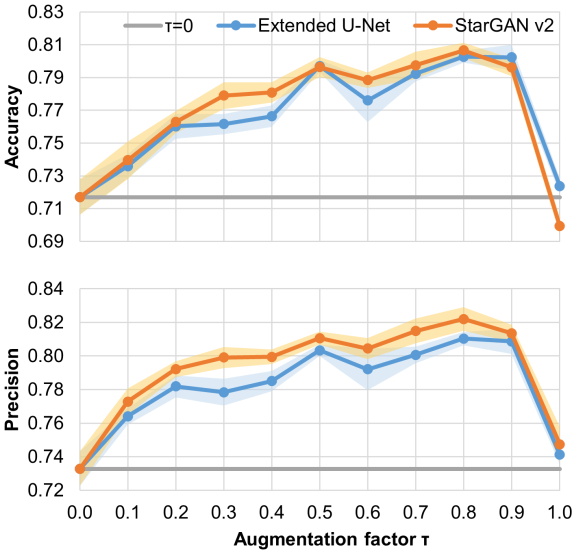

Figure 8: Accuracy and precision for different values of τ.

5.3 Evaluation of Distribution

The FID only compares datasets in terms of an estimated normal distribution over the extracted feature vectors. However, the assumption of a normal distribution is not always justified. To further investigate the feature distributions of our real and synthetic datasets, we extract for all datasets the feature vectors from an Inception v3 model trained on X^R_{train} and map them into two-dimensional space using UMAP (McInnes et al., 2020). The UMAP embedding is learned on the features of X^R_{full}. Instances are plotted in random order to prevent a biased representation due to a specific class-order within the instances.

The scatter plot of the reduced features (cf. Figure 7) reveals that, although there are notably differences between the cluster boundaries, there is a strong correlation between the feature distributions of real and synthetic data. Our synthetic images do not cover the full real data distribution, which can be partly explained by three main factors: the scarce training data, which do not cover all parts of the distribution adequately, the generator model, which possibly is not able to learn certain features from the training data, and our segmentation map generation method, which only allows to generate simple segmentation maps. The latter might also be an explanation for the unsharp cluster boundaries, as we deliberately created segmentation maps close to the class boundaries in order to generate hard samples and increase the augmentation effect of the synthetic data.

We also tried to create the same scatter plots with feature vectors extracted from the original Inception v3 model trained on a ImageNet, but UMAP failed to produce a cluster structure. This can be interpreted as a lack of information in the extracted features and corroborates our decision not to use this model for the FID.

5.4 Evaluation on Classification Use Case

We consider a test with the use case as the most important evaluation method, as the most realistic images might be worthless if they do not serve the use case and even the least realistic images might have benefits for specific applications.

With regard to our quality assurance use case, we set up a classification task based on the error classes of our training dataset. To keep it consistent with the GAN scores, we use the Inception v3 model and train it on $X^R_{\text{train,min}}$ to create a baseline. To improve the baseline and demonstrate that our synthetic images exceed the usage of vanilla image augmentations, we apply flipping, rotations, and colour augmentations during training. The training procedure, i.e. learning rate adaptation and fine-tuning, is completely automatised and kept constant through all experiments to minimise human bias on the results. The real training data is augmented by sampling a fixed fraction τ of training instances from the augmenting synthetic dataset ($X^G_{\text{U-Net}}$ or X^G_{StarGAN}). From each training run, the model with the highest accuracy on X^R_{val} is used. We measure accuracy and precision of the trained model on X^R_{test} and average the results over 10 training runs, as they slightly vary depending on parameter initialisation and training sample selection. Figure 8 summarises our results for different values of τ. Note

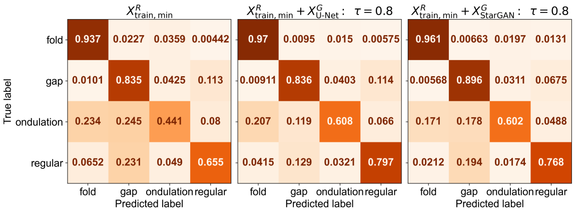

Figure 9: Confusion matrices of $X_{\text{train,min}}$ from left to right: no augmentation ($\tau = 0$), augmented with $X^G_{\text{U-Net}}$ ($\tau = 0.8$), augmented with X^G_{StarGAN} ($\tau = 0.8$).

that we calculate accuracy and precision from a confusion matrix, that is normalised over rows, to avoid bias by many instances in certain classes.

Our results confirm that our synthetic images can significantly improve the performance of a classifier when being used as data augmentation. The actual performance improvements depend less on the generator architecture as on τ. We found that for both datasets the peak performance is achieved at $\tau = 0.8$.

Moreover, the (normalised) confusion matrices show that the recall improved for every class when synthetic data augmentation was applied during training, while augmentations with $X^G_{\text{U-Net}}$ and X^G_{StarGAN} lead to comparable results despite of their different scores in 5.2 (cf. Figure 9). We find that the confusion matrix is the better alternative to the inception scores in order to evaluate class-conditional GANs, as it takes into account the correct/conditional class of an image and breaks down the scores of results within the different classes.

6 CONCLUSION

In this work we showed that GANs can effectively be used to generate synthetic FRP images for the purpose of data augmentation in a data scarce environment. In our implementation we adapted BicycleGAN and tested two generator architectures, namely Extended U-Net and StarGAN v2. We were able to achieve the goal of generating realistic textile images with both architectures.

The generated images in this work are thoroughly validated in four different ways. Firstly, a qualitative visual inspection was performed to judge the realism of the generated images. Secondly, two scores were implemented, namely IS and FID with their respective class-conditional versions, to add an objective quantitative inspection. Thirdly, the distribution of original and synthetic images were compared in terms of their feature clustering structure using UMAP. It was found that the synthesised images share a high overlap with the real images, but do not cover the distribution of the real images and are therefore less varied. Lastly, a classifier was trained to predict the type of error found in the images, where using a mix of real and synthesised data significantly improved the performance of the classification model.

In future work, we plan to extend our approach to three-dimensional weaves. This would emphasise the effects of certain defects, such as ondulations and folds, which would require either 3D segmentation maps, or a combination of 2D segmentation from different angles. The output of such model could be 3D textile objects. Furthermore, we want to explore disentanglement learning methods for generative models to remove the necessity of segmentation maps for a controlled synthesis of FRP images.

With our contribution we show that generative adversarial networks are usable in an industrial use case of manufacturing FRP. They are capable to learn high-level feature representations of the observation from which they synthesise new realistic and useful images. In order to further investigate the learned representations in the future, we believe that feature disentanglement methods may unravel meaningful knowledge about the underlying manufacturing process.

ACKNOWLEDGEMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC-2023 Internet of Production - 390621612. Simulations were performed with computing resources granted by RWTH Aachen University under project thes0947.

REFERENCES

Barratt, S. and Sharma, R. (2018). A Note on the Inception Score.

Benny, Y., Galanti, T., Benaim, S., and Wolf, L. (2021). Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731.

Berger, G. and Memisevic, R. (2017). Incorporating long-range consistency in cnn-based texture generation. *ArXiv*, abs/1606.01286.

Choi, Y., Uh, Y., Yoo, J., and Ha, J. W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Piscataway, NJ. IEEE.

Fayyaz, R. A., Maqbool, M., and Hanif, M. (2020). Textile design generation using gans. In *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–5.

Fuhr, J.-P. (2017). *Schichtbasierte Modellierung von Fertigungseffekten in der Struktursimulation von Faserverbundwerkstoffen*. Verlag Dr. Hut.

Gatys, L., Ecker, A. S., and Bethge, M. (2015). Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*, 28.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637. Curran Associates, Inc.

Jetchev, N., Bergmann, U., and Vollgraf, R. (2017). Texture synthesis with spatial generative adversarial networks.

Jing, J.-F., Ma, H., and Zhang, H.-H. (2019). Automatic fabric defect detection using a deep convolutional neural network. *Coloration Technology*, 135(3):213–223.

K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Liu, G., Gousseau, Y., and Xia, G.-S. (2016). Texture synthesis through convolutional neural networks and spectrum constraints. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3234–3239.

McInnes, L., Healy, J., and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

Miao, M. and Xin, J. H. (2018). Engineering of high-performance textiles. The Textile Institute Book Series. Woodhead Publishing.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical image computing and computer-assisted intervention - MICCAI 2015*, Lecture notes in computer science, pages 234–241, Cham and Heidelberg and New York and Dordrecht and London. Springer.

S. Schreiber, J. Geldenhuys, and H. de Villiers (2016). Texture synthesis using convolutional neural networks with long-range consistency and spectral constraints. In *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 1–6.

Sacco, C., Radwan, A. B., Beatty, T., and Harik, R. (2019). Machine learning based afp inspection: A tool for characterization and integration. In *SAMPE 2019 Conference & Exhibition. Charlotte, NC: SAMPE. doi*, volume 10, pages 19–1594.

Sacco, C., Radwan, A. B., Harik, R., and Van Tooren, M. (2018). Automated fiber placement defects: Automated inspection and characterization. In *Proceedings of the SAMPE 2018 Conference and Exhibition, Long Beach, CA, USA*, pages 21–24.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved Techniques for Training GANs. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 2234–2242. Curran Associates, Inc.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Türe, Y. and Türe, C. (2020). An assessment of using aluminum and magnesium on co2 emission in european passenger cars. *Journal of Cleaner Production*, 247:119120.

Wei, B., Hao, K., Tang, X.-s., and Ding, Y. (2019). A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes. *Textile Research Journal*, 89(17):3539–3555.

Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., and Hays, J. (2018). Texturegan: Controlling deep image synthesis with texture patches. pages 8456–8465.

Yi, X., Walia, E., and Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:101552.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017). Toward multimodal image-to-image translation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 465–476. Curran Associates, Inc.

Zhu, X., Liu, Y., Li, J., Wan, T., and Qin, Z. (2018). Emotion classification with data augmentation using generative adversarial networks. In Phung, D., Tseng, V. S., Webb, G. I., Ho, B., Ganji, M., and Rashidi, L., editors, *Advances in Knowledge Discovery and Data Mining*, Lecture notes in computer science, pages 349–360, Cham. Springer International Publishing.