Uni and Multimodal Data Augmentation using Generative Adversarial Networks for Enhanced Multi Failure Classification in Turbine Engine Blades

By

Paul IJzermans

In partial fulfilment of the requirements for the degree of
Master of Science
at Delft University of Technology,
to be defended publicly on *16-09-2024*

Faculty:	Aerospace Engineering
Department:	Control and Simulations
Programme:	Air Traffic Operations

Mentors / Supervisors:	Marcia Baptista (moved to Portugal, replaced by Ingeborg de Pater)
Graduation committee:	Alessandro Bombelli Roberto Merino Martinez Ingeborg de Pater Marcia Baptista (external advisor) Madelyn Deuten (NLR)

Keywords:
HPT-Blade
Generative Adverarial Networks
RGBD
GAN-based Augmentation
Turbine Engine Maintenance
Deep Learning
Generative Deep Learning

Abstract

Inherent subjectivity, inefficiencies, and the substantial cost related to human-based visual inspection of high-pressure turbine (HPT) blades has driven research in alternative automated techniques. The combination of computer vision (CV) and deep learning (DL) provides a compelling alternative. However, as DL models increase in capability, their large parameter spaces require substantial data toachieve robust training. Therefore, this study explores the use of two Auxiliary Classifier Generative Adversarial Networks (AC-GANs) to augment proprietary datasets for two failure modes: (a) a 3-channel red-green-blue (RGB) model for obstructed holes, and (b) a 4-channel RGB plus depth (RGBD) model for foreign object damage, with the depth reconstructed using monocular depth estimation. A Differential Evolution Optimizer (DEO) was used for hyperparameter optimization of a ResNet-18 classification target model. In the obstructed hole dataset, GAN-based augmentation showed significantly improved accuracy, recall, F1, and AUC-ROC ($p < 0.05$), competing with traditional methods using less augmentation. In the foreign object damage dataset, the inclusion of depth information significantly enhanced accuracy, precision, F1, and AUC-ROC performance ($p <0.05$). However, the RGBD augmentation mainly resulted in a trade-off between precision and recall, without statistically significant differences ($p > 0.05$)

Uni and Multimodal Data Augmentation using Generative Adversarial Networks for Enhanced Multi Failure Classification in Turbine Engine Blades

P.M. IJzermans

[a]Technical University of Delft, The Netherlands, Kluyverweg 1, Delft, 2629 HS, The Netherlands
[c]Royal Netheralands Aerospace Centre, Anthony Fokkerweg 2, Amsterdam, 1059 CM, ,

ARTICLE INFO

Keywords:
High Pressure Turbine (HPT) Blade Inspection; Deep Learning; Generative Deep Learning; Auxiliary Classifier Generative Adversarial Networks; RGB(D) Augmentation; Automated Turbine Blade Inspection

ABSTRACT

Inherent subjectivity, inefficiencies, and the substantial cost related to human-based visual inspection of high-pressure turbine (HPT) blades has driven research in alternative automated techniques. The combination of computer vision (CV) and deep learning (DL) provides a compelling alternative. However, as DL models increase in capability, their large parameter spaces require substantial data to achieve robust training. Therefore, this study explores the use of two Auxiliary Classifier Generative Adversarial Networks (AC-GANs) to augment proprietary datasets for two failure modes: (a) a 3-channel red-green-blue (RGB) model for obstructed holes, and (b) a 4-channel RGB plus depth (RGBD) model for foreign object damage, with the depth reconstructed using monocular depth estimation. A Differential Evolution Optimizer (DEO) was used for hyperparameter optimization of a ResNet-18 classification target model. In the obstructed hole dataset, GAN-based augmentation showed significantly improved accuracy, recall, F1, and AUC-ROC ($p < 0.05$), competing with traditional methods using less augmentation. In the foreign object damage dataset, the inclusion of depth information significantly enhanced accuracy, precision, F1, and AUC-ROC performance ($p < 0.05$). However, the RGBD augmentation mainly resulted in a trade-off between precision and recall, without statistically significant differences ($p > 0.05$).

1. Introduction

Air transport has played an important role in shaping global transportation, trade and tourism, and has been a driving force for local economies. The sector's growth, supported by historical demand increase (Franz, Rottoli and Bertram, 2022) and a demonstrated resilience in the post-COVID era (Sun, Wandelt and Zhang, 2023), is expected to exert increasing pressure on aircraft maintenance operations. Maintaining the highest levels of reliability and safety is crucial, particularly when it comes to critical components like turbine engines.

Turbine engines function according to the fundamental principles of gas turbine operation, which encompass compression, combustion, and expansion to convert chemical energy into mechanical energy. In this process, turbine engine blades are exposed to extreme environmental factors such as mechanical loadings, high pressures and operating temperatures, and high velocity alien particles (Aust, Shankland, Pons, Mukundan and Mitrovic, 2021; Juarez, Gutierrez and Petersen, 2023). To ensure the safety, reliability, and longevity of turbine engine blades, Reactive Maintenance (RM), Preventative Maintenance (PM), and Predictive Maintenance (PdM) strategies are deployed. These strategies aim to protect the blades, mitigate downstream engine repercussions, and minimize the operational downtime.

As part of these maintenance strategies, various inspection procedures can be initiated based on the state and conditions of the turbine engine. Typically, initial procedures involve Non-Destructive Testing (NDT) methods to minimize impact and duration. This can include bore-scope inspection using a specialized rigid camera and may extend to more complex NDTs such as Magnetic Particle Testing (MPT) for surface irregularities (Uludag, 2016), Acoustic Emission Testing (AET) (Zhang, Yang and Hu, 2018), or Infrared Thermography Testing (ITT). Despite advancements in NDT methods, human-based visual inspection of turbine engine blades remains common. Engineers conduct piece-part visual assessment of the blades to ensure they meet performance and safety standards. However, these inspections are often *subjective*, *inefficient*, and *costly*. Given that one in three accidents and one in four fatalities are linked to maintenance practices (Marais and Robichaud, 2012; Maddox)—of which an estimated 80% are due to human factors—structural failures, particularly in the engine (Aust and Pons, 2022), are recognized as a leading cause of maintenance-related incidents.

In response to these challenges, research into automated inspection using Machine Learning (ML), part of Artificial Intelligence (AI), in combination with Computer Vision (CV), has gained interest. ML techniques allow machines to learn without explicit programming, typically using vast amounts of training data.
More notably, Deep Learning (DL), a specialized subset of ML, has the ability to autonomously extract complex structures and hierarchical features by employing a sequence of mathematical layers commonly known as Deep Neural Networks (DNNs). In the context of visual inspection, Convolutional Neural Networks (CNNs), a special type of

DNN, are the de-facto models for high-dimensional image-based tasks as they allow for efficient parameter usage (amongst other characteristics such as spatial invariance). The adoption of CV and DL could potentially improve the quality, and therefore, safety of turbine blade inspection by assisting or (eventually) replacing human-based inspection.

Integrating DL with CV offers considerable promise, but a fundamental limitation is the requirement for large training datasets (Aust et al., 2021). Optimizing the parameter space relies heavily on the availability of training data, which is particularly problematic for turbine engine blade failure modes as training data can be *scarce* and *costly* to obtain. Training on inadequate datasets can cause overfitting causing reduced generalization capabilities, adversely affecting model performance (Pandey, Singh and Tian, 2020; Antoniou, Storkey and Edwards, 2017). To address this, different approaches can be utilized, including regularization techniques, few-shot learning, and transfer learning. Another widely used technique is *augmentation*, where small modifications are introduced in the original dataset using geometric or photometric transformations to expand the dataset. While these traditional augmentation methods have been proven to be effective (Khosla and Saini, 2020; Hussain, Gimenez, Yi and Rubin, 2017), they are constrained by their limited ability to generate variability.

This limitation has led to the recent development of enhanced augmentation techniques using *Deep Generative Models* (DGMs). Particularly interesting is a specific type of DGM known as *Generative Adversarial Networks* (GANs) due to its unique *implicit* learning ability (Section 2.3). These models consist of two DNNs — a generator and a discriminator — competing in an adversarial mini-max game. In a GAN, the generator has the objective of fooling the discriminator into believing the data is real, while the discriminator has the objective of correctly identifying whether the data is real or generated. This competition results in the generator approximating the underlying data distribution of the original dataset. More specifically, this research utilizes a variant known as the *Auxiliary Classifier GAN (AC-GAN)*, which incorporates an additional loss term for class-specific training. The approximated distribution can then be used to synthesize new data to augment the original dataset.

Two main unexplored areas have been identified, which form the basis for this study's contributions. The first area is the application of GANs to real-world application in turbine engine blade failure modes, specifically *obstructed holes* and *foreign object damage*. Secondly, is the use of a multimodal AC-GAN for red-green-blue-depth (RGBD) using 4-channels to integrate geometric information. This leaves a research gap with potential value in the particular domain of turbine engine blade inspection but also in more general (maintenance) applications. Based on the problem statement, as well as these identified research gaps, the research question is formulated as follows:

How Does Uni and Multimodal Data Augmentation using Generative Adversarial Networks (GANs) Affect the Performance of Multi Failure Classification in High-pressure Turbine Engine Blades?

As part of a broader initiative of the *Royal Netherlands Aerospace Centre* (NLR) supported by *Royal Dutch Airlines* (KLM) focused on automated HPT inspection, the primary objective of this research is to enhance classification performance of a downstream target model by utilizing GAN-synthesized data. This includes improving performance in both the RGB domain in the obstructed hole dataset and the RGBD domain in the foreign object damage dataset. The proprietary data used stems from real turbine engine blades with artificially applied failure modes.

This study contributes to both academic and industrial knowledge in the following ways:

i) Developed a large-scale AC-GAN for RGB data augmentation, significantly enhancing accuracy, recall, F1, and AUC-ROC scores in obstructed hole classification; *ii)* First to propose the application of a monocular depth estimator, specifically the Multiple Depth Estimation Accuracy with Single Network (MiDaS), for enhanced foreign object damage classification, resulting in significantly enhanced accuracy, precision, F1, and AUC-ROC; and *iii)* Proposed a novel 4-channel red-green-blue-depth (RGBD) AC-GAN for augmentation purposes, effectively resulting in a trade-off in recall and precision without statistically significance.

The following research paper is structured as follows:

Related Work (Section 2) provides a lay of the theoretical landscape of DGMs, to subsequently scope down on GANs for augmentation purposes. In Case Study (Section 3), a description on the case study specific stage-1 high-pressure turbine (HPT) blades will be provided including the failure modes of interest. The Methodology (Section 4) outlines the established data pipeline, covering data acquisition, preprocessing, model training, and data synthesis. It details the nature of the collected data, the custom models used, their architecture, and the training procedures. In the Results and Discussion (Section 5), the statistical analysis is briefly summarized, followed by an examination of baseline performance, augmentation effects, and a comparison with traditional methods. The implications are then discussed. In Conclusion (Section 6), an overview of the key findings will be provided referring back to the main research question. Limitations and Future Work (Section 7) addresses the constraints encountered in this study and outlines potential directions for future research.

Further explanation on the technical setup, along with verification, validation, performance metrics and statistical analysis, is provided in the Supporting Work due to their extensive nature.

2. Related Work

The following section covers the general concept of the CNNs used in both the target model and GANs, offers a general overview of DGMs, and then focuses on the underlying working mechanism of GANs. It also highlights key studies where GANs have been applied for data augmentation.

2.1. Convolutional Neural Networks

Ever since the paper *"ImageNet Classification with Deep Convolutional Neural Networks"* (Krizhevsky, Sutskever and Hinton, 2012) presented at the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), CNNs have become the de facto model for classification tasks of images. The concept of CNNs is inspired by the neural connections within the visual cortex, where individual neurons respond to inputs within specific regions known as receptive fields. These overlapping receptive fields together make the entire visual area (Lindsay, 2021). To accomplish this, CNNs make use of two special layers known as *convolutional layers* and *pooling layers*.

Convolutional layers create feature maps from input images by applying kernels. Each input channel (e.g., RGB) has trainable kernels that detect specific features (e.g. edges). The kernel dimensions and stride determine how features are merged to form the output feature map, significantly reducing the number of parameters (compared to fully connected layers) and ensuring consistent processing across the image. This method eliminates the need to recognize the same object at every location, crucial for extracting detailed hierarchical features and achieving efficient local-to-global processing through multiple layers (Prince, 2023).

Pooling layers trim the spatial dimensions (known as down-sampling) of a feature map while preserving crucial details. This process is essential for several reasons. In high-dimensional data (such as images) pooling layers reduce the parameter load. Furthermore, a convolutional layer's output is sensitive to the specific receptive field of each pixel. This sensitivity means that even minor shifts in the data can alter the output. Down-sampling, enhances the network's resilience to variations in feature position and location within the input data, known as translation invariance.

In combination with conventional layers convolutional and pooling layers are fundamental components of CNN-backbone architectures for effective image processing. In both academic and industrial contexts, models like Inception, Visual Geometry Group (VGG), and Residual Network (ResNet) are frequently used, with ResNet playing a central role in this research (Section 4.3).

Moreover, both convolutional layers and their inverse counterparts, deconvolutional layers, are crucial components of the developed AC-GAN (Section 4.5) used for augmentation.

2.2. Deep Generative Models

Deep Generative Models (DGMs), part of Generative AI, lay at the intersection of Generative Models and DNNs. Generative Models aim to to capture the underlying data distribution of the data, enabling them to perform a broad range of downstream tasks, including synthetic data generation. Whereas discriminative models are designed to learn the conditional probability $p(y|x)$ of a label y given a set of features x, they do not require the modelling of the distribution of the features themselves. discriminative models use the concept of decision boundaries in the feature space, which can be a simple line or a more complex manifold, to separate classes. While discriminative models can classify or differentiate objects by identifying a few salient patterns, DGMs tackle a more difficult task. DGMs are tasked with modeling the entire data distribution, which requires capturing a broad range of correlations and dependencies such as spatial relationships and co-occurrences within the data. This ability of DGMs is highly valuable in specific domains and application such as high-resolution image synthesis (Karras, Aila, Laine and Lehtinen, 2017), super-resolution (Liu, Siu and Chan, 2020), text-to-image conversion (Li, Qi, Lukasiewicz and Torr, 2019a), and image-to-image translation (Li, Tang, Zhang, Zhang, Li and Yan, 2019b). However, the applicability is not limited to imagery. DGMs are increasingly used in video processing (He, Yang, Zhang, Shan and Chen, 2022; Ho, Chan, Saharia, Whang, Gao, Gritsenko, Kingma, Poole, Norouzi, Fleet et al., 2022) and audio (Oord, Dieleman, Zen, Simonyan, Vinyals, Graves, Kalchbrenner, Senior and Kavukcuoglu, 2016).

Different types of (hybrid) DGMs exist and are continuously being developed. Figure 1 presents a taxonomy of the more well-known (non-hybrid) types of DGMs, categorized based on different (mathematical) principles. GANs are part of the latent variable models, more particularly the implicit models. Unlike explicit models that require explicit density estimation, implicit models, such as GANs, avoid this complexity by not modeling it directly. This approach makes GANs particularly effective for tasks like image generation, where modeling complex density functions can be challenging.

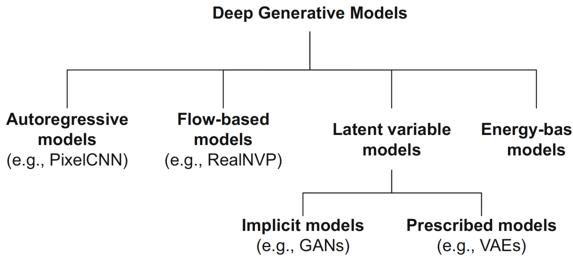

Deep Generative Models
Autoregressive models (e.g., PixelCNN)
Flow-based models (e.g., RealNVP)
Latent variable models
Energy-based models
Implicit models (e.g., GANs)
Prescribed models (e.g., VAEs)

Figure 1: Taxonomy of Deep Generative Models (DGMs), classifying various methods based on their underlying principles. GANs are categorized under implicit models within the broader class of latent variable models (Tomczak, 2022).

2.3. Generative Adversarial Networks

The concept of GANs, first introduced by Ian Goodfellow et al. in their seminal paper "Generative Adversarial Nets" (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville and Bengio, 2014) in 2014 (further referred to as *original* GAN), represent a unique generative framework characterized by the training approach. GANs are a type of latent variable model. Similar to other DGMs, they have the objective of solving the generative modelling problem of observing a collection of data points, and learn their probability distribution $P(x)$. In a GAN, as can be seen in Figure 2, two distinct networks are present with opposing tasks, namely a generator and discriminator.

The generator samples from a noise distribution and maps it to the approximated data distribution (x'). The discriminator receives samples from both the approximated data distribution (x') and the real data distribution (x). The discriminator is trained to distinguish between real and generated samples, while the generator is trained to fool the discriminator.

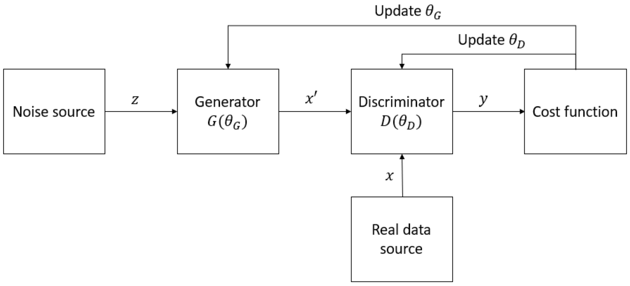


Figure 2: Schematic of the Generative Adversarial Network (GAN) framework, illustrating the interaction between the generator and discriminator during the training process.

Where explicit DGMs typically rely on (approximate) Maximum Likelihood Estimation (MLE), GANs utilize a distinct loss function. As shown in Formula 1, the loss function, denoted by $\mathcal{L}$, represents the adversarial interaction between two networks: the generator G and the discriminator D. The generator G is trained to minimize $\mathcal{L}$ with respect to its parameters θ_G, while the discriminator D is trained to maximize $\mathcal{L}$ with respect to its parameters θ_D. Here, $D(x; \theta_D)$ represents the discriminator's ability, using parameters θ_D, to correctly classify real data points x. Conversely, $G(z; \theta_G)$ denotes the data generated by the generator, parameterized by θ_G, based on samples z drawn from the latent space.

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D} \mathcal{L}(D, G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x; \theta_D)] \\ + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z; \theta_G); \theta_D))]\end{aligned} \tag{1}$$

Initially, both networks start with randomized parameters. First, the discriminator undergoes training, evaluating both real and fake samples. At this stage, fake samples are essentially random noise because the generator is untrained. The parameters θ_D are updated using a batch size N, real data samples $x^{(i)}$, and latent input $z^{(i)}$ by using the loss function from Equation 1 and calculating the gradient with respect to the θ_D, as shown in Equation 2, keeping θ_G constant.

$$\begin{aligned}\nabla_{\theta_\text{D}}\mathcal{L}_\text{D} = -\frac{1}{N}\sum_{i=1}^{N}\Bigg[\frac{1}{D(x^{(i)})}\nabla_{\theta_\text{D}} D(x^{(i)}) \\ -\frac{1}{1 - D(G(z^{(i)}))}\nabla_{\theta_\text{D}} D(G(z^{(i)}))\Bigg]\end{aligned} \tag{2}$$

Following the discriminator's training, the training process shifts towards optimizing the generator while keeping θ_D constant. The generator employs feed-forward mapping from the latent space to produce fake data, and its performance is measured using the discriminators ability. Similarly, with batch size N and data points $G(z^{(i)})$, the gradients with respect to θ_G are computed using the loss function from Equation 1 and taking the derivative with regards to θ_G, as can be seen in Equation 3.

$$\nabla_{\theta_\text{G}}\mathcal{L}_\text{G} = \frac{1}{N}\sum_{i=1}^{N}\nabla_{\theta_\text{G}}\left[\log(1 - D(G(z^{(i)})))\right] \tag{3}$$

Different discriminator loss functions and optimizers can be employed in GANs. However, the original GAN framework utilizes Binary Cross-Entropy (BCE) as the discriminator loss function and the Momentum algorithm to enhance Stochastic Gradient Descent (SGD). The Momentum algorithm introduces a velocity term, which helps the optimizer navigate through local minima more effectively.

Using this learning process, a GAN is able to implicitly learn the underlying probability distribution of the training data. In Figure 3, a more pedagogical explanation is provided. Here it can be seen that GANs are trained by simultaneously updating the discriminative distribution (blue, dashed line) so that it discriminates between samples from the real distribution (black, dotted line) $P(x)$ and the generative distribution $P(x')$ (green, solid line). The lower horizontal line is the domain from which z is sampled, in this case uniformly. The horizontal line above is part of the domain of x' with the mapping $x' = G(z)$ (Goodfellow et al., 2014). It can be observed that the green line slowly converges to the same distribution as the real distribution.

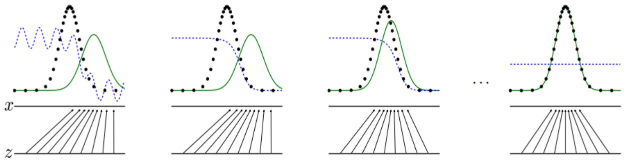


Figure 3: Pedagogical explanation of the training procedure of a Generative Adversarial Network training principle (Goodfellow et al., 2014).

2.4. Generative Adversarial Network Variants

The original GAN framework, as can be seen in Figure 4 (a), has undergone significant enhancements since its inception. In the context of the image generation, two developments have been particularly noteworthy.

Firstly, the incorporation of deep convolutional and deconvolutional layers, known as Deep Convolutional GANs (DC-GANs), has significantly improved image processing capabilities. CNN-based models enhance the efficiency and robustness of image data handling through the specialized layers. For simplicity, the GANs referenced in this research are all standard deep convolutional GANs and will henceforth be referred to simply as GANs.

Secondly, the introduction of conditional GANs (cGANs) in *"Conditional generative adversarial nets"*, as can be seen in Figure 4 (b)(Mirza and Osindero, 2014) and, subsequently, Auxiliary Classifier GANs (AC-GANs) in "*Conditional image synthesis with auxiliary classifier gans*" (Odena, Olah and Shlens, 2017), as can be seen in Figure 4 (c) allowed for a level of controllability of the output using a conditional variable in the generation process. This variable can encompass a wide range of information, such as class labels or data from different modalities. While a cGAN conditions the generator and discriminator, the AC-GAN does not condition the classifier but adds a classification loss. This means the loss back propagated for both the generator and discriminator includes a component for class conformity.

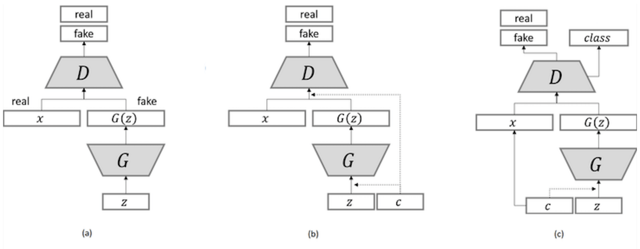


Figure 4: Different GAN variants with the original GAN (a), a cGAN (b) and an AC-GAN (c) (Zhan et al., 2023).

Given the central role of AC-GANs in this research, it is important to explore their functionality in greater detail. The original GAN loss function, referred to as the *adversarial loss* and presented in the previous section (Equation 1), is expanded in AC-GANs to include additional class-specific loss terms, known as the *classification loss*.

To adjust the discriminator, an additional term is added to the loss function, capturing the class prediction probability $D_{\text{class}}(c_x \mid x)$ from the pair (x, c_x) sampled from the real data distribution $p_{\text{data}}(x, c_x)$, as shown in Equation 4. This term ensures that the discriminator not only differentiates between real and fake samples but also correctly classifies the real data into the appropriate class c_x.

For the generator, the loss function is similarly augmented to account for class information. Specifically, the generator's objective includes $D_{\text{class}}(c_z \mid G(z))$, where z is drawn from the noise distribution $p_z(z)$ and c_z from the label distribution. This term, also reflected in Equation 4, encourages the generator to produce data that not only appears realistic but is also consistent with the specified class labels.

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D}\mathcal{L}(D,G) = {} & \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x;\theta_D)\right] \\ & + \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D_{\text{class}}(c_x \mid x;\theta_D)\right] \\ & + \mathbb{E}_{z\sim p_z(z)}\left[\log(1 - D(G(z;\theta_G);\theta_D))\right] \\ & + \mathbb{E}_{z\sim p_z(z)}\left[\log D_{\text{class}}(c_z \mid G(z;\theta_G);\theta_D)\right] \end{aligned} \tag{4}$$

A more detailed explanation of the code translation is provided using pseudo-code in Algorithm 1 (Appendix A).

2.5. Data Augmentation using Generative Adversarial Networks

The concept of *GAN-based augmentation* is similar to traditional augmentation. Traditional augmentation is a widely established and recognized element in ML pipelines where data is slightly manipulated to inject variability for training. Broadly, these methods can be divided into two categories, namely geometric and photometric augmentation.

Geometric transformations involve subtle alterations to image geometry, such as cropping, resizing, cutout, and rotation, as illustrated in Figure 5 (b). Additionally, these transformations encompass non-linear warping methods, offering a wide range of techniques for image modification. These methods can be helpful for enhancing model robustness by introducing variability in the training set.

Photometric methods pertain to the transformation of pixel intensities (Taylor and Nitschke, 2018). Unlike geometric transformations, they do not alter the geometry characteristics of an image. Examples of color transformations include Gaussian noise and jitter color, as can be seen in Figure 5 (c). In essence, these methods involve random adjustments within image color spaces.

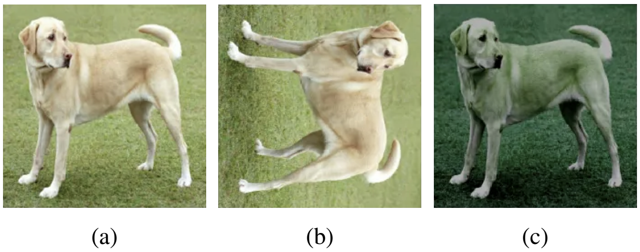


Figure 5: Normal images (a) and traditional augmentation methods including rotation (b) and jitter color distortion (c)

GAN-based augmentation leverages the generative capabilities of GANs to create new, diverse data, thereby increasing variability. This approach can help promote better generalization and reduce overfitting. GAN-based augmentation can be classified into three key subdomains: *unsupervised*, *semi-supervised*, and *supervised* augmentation.

Unsupervised GAN-based augmentation involves using GANs trained without *labeled* data. A classic example is the original GAN, which generates images without targeting specific attributes or classes. While this approach lacks control over the characteristics of the generated outputs, it can still enhance a dataset, provided it generates data within a single class. However, with the advent of more advanced models such as cGAN and AC-GAN, which allow for control over the generation of multi-class data, unsupervised GAN-based augmentation has become less prevalent in multi-class augmentation practices.

Semi-supervised GAN-based augmentation is practical in scenarios where there is ample data but a scarcity of *labeled* examples. This method leverages the abundance of unlabeled data to better capture the inherent structure of the data distribution (Madani, Moradi, Karargyris and Syeda-Mahmood, 2018b). The GAN is trained with both labeled and unlabeled data to introduce additional variability. A common practice in this domain is using the discriminator as the final target model, as the GAN training process enables it to learn from labeled, unlabeled, and synthetic data.

Supervised GAN-based augmentation will be utilized in this research. This approach uses labeled data to guide the GANs in generating class-specific synthetic samples. Widely used models in this domain include the cGAN and AC-GAN. It is important to note, however, that training GANs can be notoriously challenging due to their inherent mini-max adversarial game. Ensuring stable training can be difficult, with issues such as mode collapse, imbalanced generator and discriminator power, and the tendency to either overfit or underfit the data (Ahmad, Jaffri, Chen and Bao, 2024). Although promising, to our knowledge, there has been limited application of GAN-based augmentation for enhanced failure classification, particularly in turbine engine blades. However, other fields have demonstrated successful implementations, providing strong evidence of its potential for broader adoption.

Demonstrative use of supervised GAN augmentation exists. For example, in "A Low Shot Learning Method for Tea Leaf's Disease Identification" (Hu, Wu, Zhang and Wan, 2019) it was demonstrated that a significant improvement in disease identification in tea leaves using DC-GAN augmented samples, leading to an average accuracy increase of 28% over traditional augmentation methods with a VGG16 target model, as discussed in *"A Low Shot Learning Method for Tea Leaf's Disease Identification"*.

In the study *"Tomato Plant Disease Detection Using Transfer Learning with C-GAN Synthetic Images"*, (Abbas, Jain, Gour and Vankudothu, 2021) a DC-GAN was deployed to generate synthetic images of tomato plants. Here, a pre-trained DenseNet121 which had the highest accuracy among target models VGG19, ResNET50, Inception-V3, Xception, MobiNet, Densenet169, and DenseNet 201 was used. The model achieved an accuracy of 98.16%, 95.08%, 94.34%, on the original PlantVillage dataset for 5-class classification, 7-class, and 10-class classification tasks, respectively, and it achieves an accuracy of 99.51%, 98.65%, 97.11% with the original PlantVillage plus synthetic images dataset for 5-class classification, 7-class, and 10-class classification tasks, respectively. Here, the deployment of GAN-based data augmentation has yielded positive outcomes.

Another study titled *"Chest X-Ray Generation and Data Augmentation for Cardiovascular Abnormality Classification"* (Madani, Moradi, Karargyris and Syeda-Mahmood, 2018a), employed a dataset containing 2,134 normal and 1,976 abnormal frontal chest X-rays. The study reported an initial classification accuracy of 81.93% without augmentation, which increased to 83.12% upon the application of conventional augmentation techniques. With the integration of GAN-based augmentation, generating 500 synthetic images for each class, the accuracy further increased to 84.19%.

Further evidence of the potential efficacy of GANs as augmentation method is found in the work *"GAN-based Synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification"* (Frid-Adar, Diamant, Klang, Amitai, Goldberger and Greenspan, 2018). Their research delineated the superiority of DCGAN- augmented datasets in enhancing the diagnostic precision of a non-pretrained custom CNN. Post GAN-augmentation (CNN-AUG-GAN), a marked increase was observed in both sensitivity, rising from 78.6% to 85.7%, and specificity, from 88.4% to 92.4%. This contrasted with the performance improvements seen with traditional augmentation methods (CNN-AUG), substantiating the potential of GANs in improving performance in CNN-based applications for medical classification tasks.

It should be noted, that GAN-based augmentation is not guaranteed to be an effective method and is contingent upon several factors, including the baseline performance of the target model, the quantity and quality of the data, the complexity of the data, and the configurations of the GAN itself. For example, in the study titled *"Data Augmentation Using Generative Adversarial Networks (GANs) for GAN-based Detection of Pneumonia and COVID-19 in Chest X-Ray Images"* (Motamed, Rogalla and Khalvati, 2021), the efficacy of GANs in the domain of pneumonia and COVID-19 detection in chest X-ray imagery was evaluated. It was demonstrated that a DC-GAN failed to effectively augment data for this purpose. This suggests that the efficacy of GAN-based augmentation can vary.

3. Case Study

The following section elaborates on the case study by detailing the context of the relevant high-pressure turbine (HPT) blades, as well as elaborating on the relevant failure modes.

3.1. General Electric High-Pressure Turbine Blades

The General Electric next-generation (GEnx) turbine engine is a state-of-the-art high-bypass turbofan jet engine featuring dual rotors and axial flow. The engine is manufactured by General Electric (GE) Aerospace and consists of two models, the 1B (111-inch diameter, 70000 pounds thrust) and 2B (104-inch diameter, 67000 pounds thrust). The two variants of the engine possess a shared core but exhibit specific distinctions between models such as fan diameter, thrust rating, and aircraft compatibility. The GEnx-1B is used in the Boeing 787 Dreamliner family, which includes the 787-8, 787-9, and 787-10 variants. The GEnx-2B engine is optimized for the Boeing 747-8.

3.2. Failure Modes

The research focuses on high-pressure turbine (HPT) blades situated immediately post-combustor (stage-1), as illustrated in Figures 6a and 6b (white box). These HPT blades, of which a proximate model can be seen in Figure 6c are subjected to extreme conditions, including high temperatures and pressures, large alien objects, centripetal forces, and exposure to high-velocity airflow carrying fine particulates. Furthermore, the high temperatures contribute to the breakdown of engine lubricants, leading to the formation of sludge, varnish, and solid deposits (Juarez et al., 2023). Additionally, carbonaceous deposits and thermal degradation of fuel and airborne particles contribute to various failure modes in turbine engine blades.

Common failure modes that can impact engine function include fractures, deformations, material loss, complete blade detachment, obstructed cooling holes, cracks in the tip, platform, and airfoil, overheated areas, foreign object damage, scratches, and airfoil burns with cracks (Aust and Pons, 2019). These different failure modes result from various failure mechanisms such as fatigue, creep, corrosion, erosion, sulphidation, foreign objects impact, vibration and combinations of these mechanisms. In this research, the aim is to enhance the failure mode classification of *two* of these frequently encountered modes, namely *obstructed holes* and *foreign object damage* datasets.

Obstruction of cooling holes can pose significant risks, as these holes play a critical role in dissipating the intense heat generated during combustion. They allow compressed air from the compressor stage to flow through and create a thermal barrier on the blade surface. One common cause of this obstruction is coke formation—carbon-rich solid residues from unburned fuel. Coke formation is a major contributor to premature failure in aircraft turbine engines, leading to reduced performance and an increased likelihood of serious accidents (Wu, Zong, Fei and Ma, 2017; Kauffman, Feng and Karasek, 2000). Another frequent cause of blockage is the accumulation of small particles, such as sand and volcanic ash. This failure mode becomes especially dangerous when blade temperatures exceed the super-alloy's melting point, potentially resulting in severe consequences.

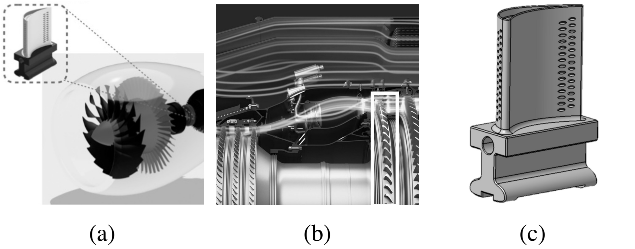


Figure 6: Component details with a) the position of the HPT blades in the turbine engine, b) a visual indication of the airflow in the engine and c) a 3D approximate model of the turbine blade.

The second failure mode of interest is the presence of damage caused by the impact of foreign objects. Foreign object damage can be caused by a variety of debris, including small stones, loose materials ingested into the engine, or even birds. These damages can result in stress concentrating at a single point, increasing the risk of failure propagation. As the material's resilience is compromised, even minor loads can exacerbate these damages, potentially leading to cracks or complete breaks (Aust et al., 2021). Another problem arises when the thermal barrier coating (TBC) is damaged, as the coating is designed to protect the material from thermal stress, oxidation, and corrosion.

4. Methodology

The following section outlines the methodology used. This methodology contributes to testing the following hypothesis:

- **Hypothesis:** Augmentation using synthetically generated data from GANs in both RGB (obstructed holes) and RGBD (foreign object damage) domains will significantly improve CNN-based performance in classifying HPT blade failures.

This hypothesis inherently requires testing another hypothesis within the research: whether the depth information derived from a cross-modal depth estimator improves performance by concatenating an additional depth channel.

The methodology, depicted in high-level workflow in Figure 7, follows the conceptual data flow and is segmented into several distinct phases: *data collection* (A), *preliminary preprocessing* (B), *target model development including hyperparameter optimization* (C), *AC-GAN development* (D), and the *classification target model evaluation* (E).

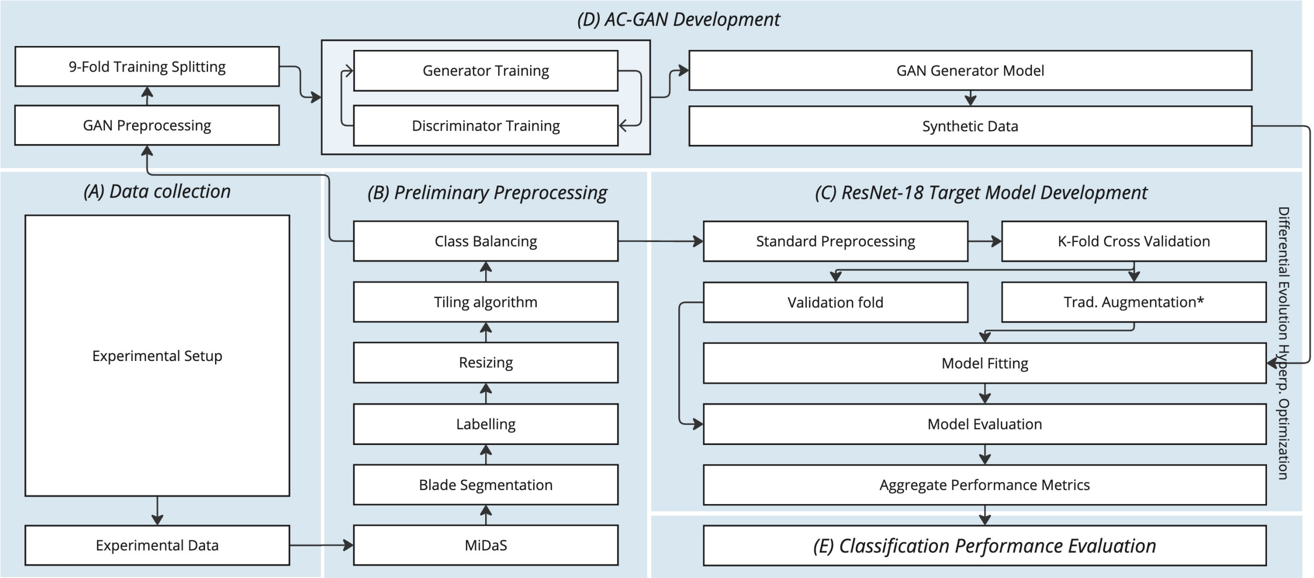


Figure 7: High-level overview of the research methodology workflow. * refers to the traditional augmentation used for comparison in the results.

4.1. Data Collection

The experimental setup was used to collect data of the two distinct failure modes. The blades, provided by KLM, displayed signs of wear such as damaged coatings but did not exhibit obstructed holes and only minor foreign object damage.

In the experimental setup, a camera was mounted on a Universal Robot robotic arm (model UR10e), as can be seen in Figures 8a and 8b with weight of 33.5 kg, max. payload of 12.5 kg and a reach of 1300 mm. The robot arm is interfaced with RoboDK software for preprogrammed arm-tip movements. a Daheng Imaging MER2-1220-9GC-P Industrial Camera connected to an Ethernet cable with a Kowa Im12fc24m lens (manual focus) was used. The camera remained constant at 15 centimeter from the turning table. For a more expansive list of the apparatus and software used, please see Supporting Work A.

To simulate the phenomenon of obstructed holes, a malleable material composed of Staedtler Fimo clay with mixed colours of "chocolate" and "nougat" was used. This material was selected for its ability to replicate the visual properties of obstructions and their associated dark coloration. An example can be seen in Figure 8c. Multiple holes in various regions of a specific blade were randomly obstructed, including the leading, concave, and trailing edges, to capture a diverse set of potential obstructions. Since only a limited number of turbine blades were available, multiple obstructions were introduced on each blade.

To simulate foreign object damage, a hammer was forcefully struck against the trailing edge of the blade, as shown in Figure 8d. Foreign object damages (including nicks, dents, and tears) are simplified here in terms of physical appearance. In practice, there is a distinction in the different forms (e.g. dents are more rounded indentations on the blade's surface, while tears involve actual fractures or splits in the material.) The impact damage mimics the effects of foreign objects striking the blade (Aust, Shankland, Pons, Mukundan and Mitrovic). Similar to the obstructed holes, multiple instances of foreign object damage were artificially applied to each blade.

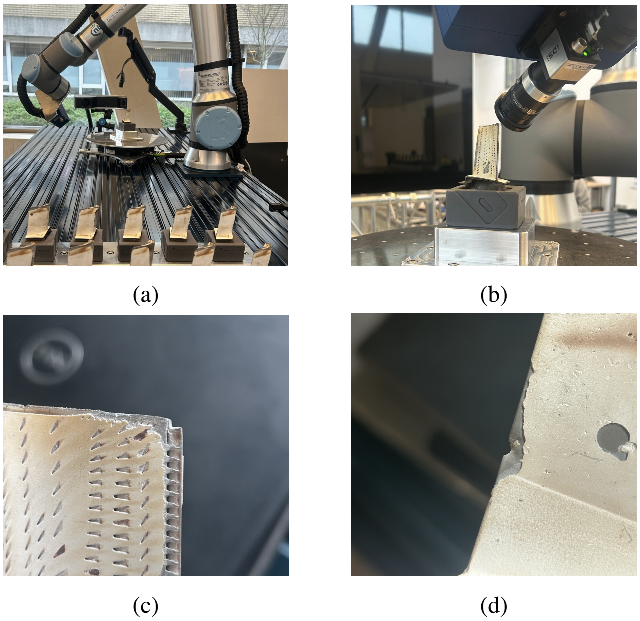

(a)
(b)
(c)
(d)

Figure 8: Overview of the data collection process with a visualization of the experimental setup with the Universal Robot arm (a), the position of the lens on the camera (b) example of an obstructed hole and (c) example of foreign object damage (d)

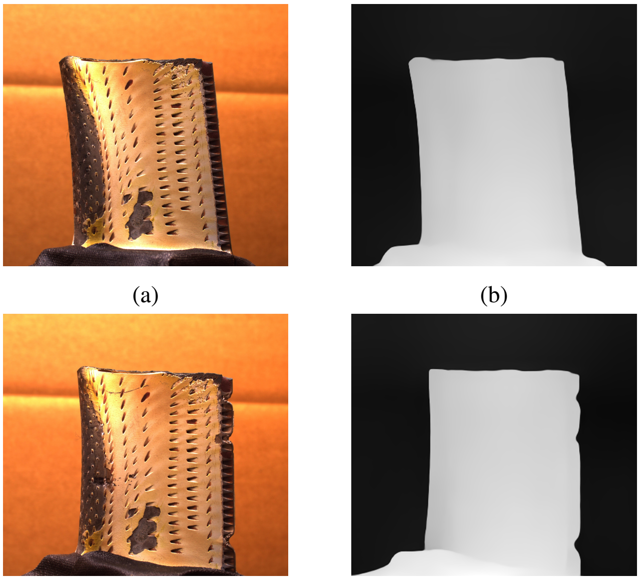
4.2. Preliminary Preprocessing

The large-scale image dimensions, as demonstrated by the raw images in Figures 9a and 9b, necessitated a series of steps to transform raw images (2977x2732 pixels), which were infeasible for data processing and AC-GAN training at this scale, into a usable format. These steps are referred to as *Preliminary Preprocessing*, a phase purposefully distinct from standard image preprocessing in ML pipelines. This phase is designed to address the specific needs of the study, encompassing cross-modal RGB-to-depth conversion, blade segmentation, labeling, resizing, tiling, and undersampling.

Cross-modal RGB-to-depth conversion was conducted using a Multiple Depth Estimation Accuracy with Single Network (MiDaS) (version 3.1) from the paper *"A Model Zoo for Robust Monocular Relative Depth Estimation"* (Birkl, Wofk and Müller, 2023a). MiDaS is a cross-modal neural net, meaning it was trained on mapping RGB to depth, and was chosen for its high accuracy in-depth estimation based on a comparative benchmark test (Birkl, Wofk and Müller, 2023b; Ranftl, Lasinger, Hafner, Schindler and Koltun, 2022; Ranftl, Bochkovskiy and Koltun, 2021). The model is trained on a diverse array of datasets (ReDWeb, DIML, Movies, MegaDepth, WSVD, TartanAir, HRWSI, ApolloScape, BlendedMVS, IRS, KITTI, NYU Depth V2) encompassing various environments and objects. Its objective is to minimize the disparity between predictions and ground truth. The resultant output is a depth map providing per-pixel depth values akin to stereo estimation. However, it's imperative to acknowledge the limitations of monocular depth estimation. The quality of the output is highly contingent upon the model's pretraining and the dataset used for this. As such, the MiDaS model was unable to map the RGB, as seen in Figure 9a, to depth with an accurate depth representation of the obstructed holes, as in Figure 9b. In the foreign object damage dataset, however, the RGB images, as exemplified in Figure 9c, were successfully used for depth creation in most angles, as shown in Figure 9d.

Blade segmentation was implemented to minimize irrelevant tiles (background). While various methods were available for this purpose, the efficacy of MiDaS was apparent. More specifically, as MiDaS is inherently equipped to differentiate between the object in the forefront and the background, and was, therefore, utilized as base for the segmentation procedure. Throughout the blade segmentation process, the image size remained unchanged to preserve quality.

Labelling was conducted using a custom algorithm designed to select each failure mode within an image using a *bounding box*. Typically, labeling an entire image suffices for classification tasks. However, due to multiple failure modes present on each blade, bounding box were used per image to select the failure mode, to subsequently use this in the tiling algorithm. A total of 750 images were labelled per failure mode.



(a) (b)

(c) (d)

Figure 9: Examples of RGB images showing obstructed holes (a) and foreign object damage (c). Additionally, MiDaS-generated depth images are presented, where the obstructed holes are not visible (b), while the foreign object damage is clearly identifiable (d).

Resizing was employed as an intermediary step to adjust the image dimensions from 2732 by 2977 pixels to 2240 by 2240 pixels. This modification allows for the use of 224-pixel tiles without cropping the image, accepting minimal loss due to interpolation.

Tiling includes the division of the resized images into smaller *tiles* of 224 by 224 pixels. This particular size was chosen, as elaborated in Section 4.3, to leverage a highly effective classification model. The algorithm processes the RGB image (10a) and the congruent depth images (in case of the foreign object damage) 10b, the labeled boxes (white square boxes) combined with the segmentation mask 10c. The tiles are then categorized and saved as *non-failure* or *failure* tiles based on a selection criterion: a 50% overlap with the selected bounding boxes for obstructed holes and a 75% overlap for foreign object damage, respectively. This threshold is acknowledged to be influential yet reasonable, approximating human judgment by engineers.
Figure 10 visualizes the process, where tiles demonstrated with a red borders are considered irrelevant, yellow-bordered tiles are disregarded due to insufficient overlap with failure boxes, green-bordered tiles are identified as failure tiles based on sufficient overlap with the bounding box, and blue-bordered tiles are used as non-failure tiles. This approach is similarly applied to foreign object damage, with the additional condition that all tiles, both failure and non-failure, need to be on the edge of the blade where the failures are located.

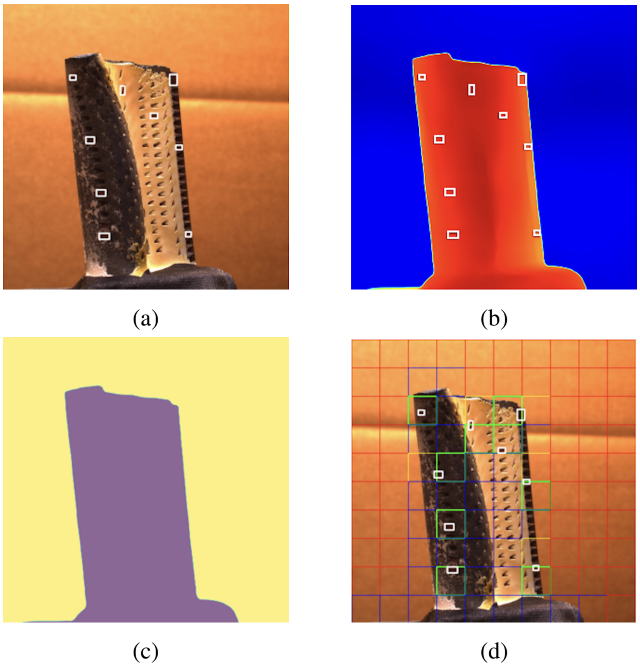


Figure 10: Intermediate visualisations of the tiling algotihm with the RGB (a) and the depth (b) image with bounding boxes, the MidaS-based segmentation mask (c) and the final tiles selection (d).

In Figure 11, examples of the resulting individual tiles with a resolution of 224x224 pixels are displayed. Sub-figures (a) and (b) show examples of non-failure tiles, while sub-figures (c) and (d) display examples of failure tiles of the obstructed holes. Sub-figures (e) and (f) present examples of non-failure tiles of the foreign object damage with their corresponding depth images (i) and (j). Additionally, sub-figures (g) and (h) illustrate failure tiles with their corresponding depth images (k) and (l).

Undersampling is used to balance the underrepresented failure tiles with the more numerous non-failure tiles. For the obstructed holes, there are 29,067 non-failure tiles compared to 3,719 failure tiles. For the foreign object damage, there are 13,558 non-failure tiles and 1,730 failure tiles. Since this research uses balanced datasets — both to ensure non-skewed performance metrics and to support more stable AC-GAN training — Table 1 provides information on the balanced datasets compared to the original output volumes of the tiling algorithm.

Table 1
Original and balanced tile counts for each failure mode with obstructed hole (OH), foreign object damage (FOD), non-failure (NF) and failure (F)

Failure Mode	Original (NF / F)	Balanced
OH	29,067 / 3,719	3,719 / 3,719
FOD	13,558 / 1,730	1,730 / 1,730

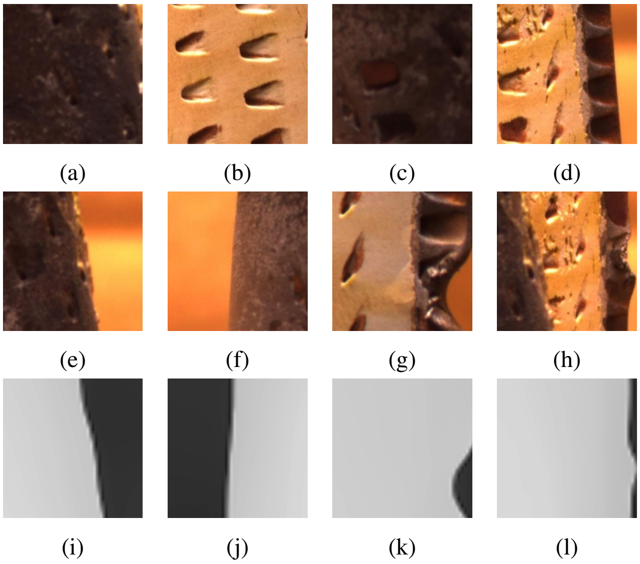


Figure 11: Examples of tiles after preliminary preprocessing: (a-d) show tiles from the obstructed holes dataset, while (e-l) show tiles from the foreign object damage dataset.

4.3. ResNet-18 based target model

Two classification models were developed using a Residual Network (ResNet) backbone for both RGB and RGBD inputs to classify non-failure and failure tiles. These models, referred to as *target models*, are used to evaluate baseline performance without augmentation, with traditional augmentation, and with GAN-based augmentation.
Additionally, these models are used to compare RGB and RGBD data to assess whether depth information improves classification in the foreign object damage dataset. The following section explains the rationale for selecting the ResNet backbone, describes its architecture, outlines the preprocessing steps, and details the modifications made to the models.

4.3.1. ResNet Architecture

A priori determination of the target model can be challenging. Whilst recognizing the different possibilities, a Residual Network-18 (ResNet-18) model, part of the ResNet-family, was chosen due to its consistent state-of-the-art performance across various image recognition benchmarks (Tan, Li, Liu, Lu and Xiao, 2020). Residual Networks (ResNet), introduced in the paper *"Deep Residual Learning for Image Recognition"* (He, Zhang, Ren and Sun, 2016), were designed to address the vanishing gradient problem (inability to propagate gradients deeper into the network). The key innovation of ResNet is the use of skip connections within "residual blocks", which enable the construction of deep networks by allowing gradients to flow more easily during backpropagation.

As can be seen in the simplified overview in Table 2, the architecture consists of an initial convolutional layer followed by four residual blocks, each containing two 3x3 convolutional layers with increasing filter sizes. The network concludes with a global average pooling layer and a fully connected layer, enabling efficient feature extraction and, in this research, classification.

Table 2
Simplified architecture of ResNet-18 used as the backbone for the target model, with key adaptations highlighted (in bold). The first convolutional layer has been modified to accept 4-channel inputs, and the final layer has been adjusted for binary classification.

Layer Name	Output Size	Description
Input Image	-	Input Image
conv1	112x112	**7x7, 64 filters, stride 2 (modified to accept 3 or 4 input channels)**
max_pool	56x56	3x3 max pooling, stride 2
conv2_x	56x56	$\begin{Bmatrix} 3x3, 64\text{filters} \\ 3x3, 64\text{filters} \end{Bmatrix} \times 2$
conv3_x	28x28	$\begin{Bmatrix} 3x3, 128\text{filters} \\ \text{stride2} \\ 3x3, 128\text{filters} \end{Bmatrix} \times 2$
conv4_x	14x14	$\begin{Bmatrix} 3x3, 256\text{filters} \\ \text{stride2} \\ 3x3, 256\text{filters} \end{Bmatrix} \times 2$
conv5_x	7x7	$\begin{Bmatrix} 3x3, 512\text{filters} \\ \text{stride2} \\ 3x3, 512\text{filters} \end{Bmatrix} \times 2$
avg_pool	-	Global average pooling
fc	-	**Fully connected, 2-d, Logits**
Total Parameters	-	Approx. 11.7M

4.3.2. Preprocessing

Several preprocessing steps are implemented to ensure input compatibility of the data with the ResNet models. A distinction is made between *basic transformations* and *traditional augmentation*.

Basic transformations are needed when using the ResNet architecture, which requires input images to be resized to 224 by 224 pixels. The RGB channels were normalized using standard ImageNet mean and standard deviation (SD) to ensure stable training. This normalization is essential for pretrained models trained on ImageNet, as these models expect input data with similar statistical properties. For non-pretrained models, these values are used as a reasonable approximation of the dataset characteristics, though they are less critical since the model can adapt to the data during training. Since standard preprocessing does not account for depth normalization in RGBD data, the mean and standard deviation for the depth channel were manually calculated.

Traditional augmentation is not strictly required but is widely used in modern ML pipelines to enhance dataset diversity through geometric and photometric transformations. These transformations are typically applied on-the-fly during data loading, allowing the model to see a slightly varied dataset each epoch. In this study, the performance of GAN-based and traditional augmentations will be compared. A geometric 15% *rotation*, a widely recognized standard technique, was found to be an effective method within the standard set of rotation, flipping, and other similar transformations. Photometric augmentation was excluded due to its negative impact on performance. This approach was consistently applied in the comparison of obstructed holes and foreign object damage datasets. Note that no optimization techniques were employed to compare all traditional methods.

4.3.3. Modifications

Architectural modifications were made to adapt the target models for *a)* binary classification, *b)* process 4-channel RGBD inputs, and *c)* allow for the implementation of Gradient-Class Activation Mapping (Grad-CAM) for the validation of the model (see Supporting Work B). A fully connected linear layer was introduced, with two logits. Cross Entropy Loss was used for binary classification on the raw logit ouputs. The choice of two logits, rather than a single logit with a sigmoid function, was made primarily to facilitate the use of Grad-CAM for validation purposes (See Supporting Work B). This technique allows for class-specific backpropagation and activation mapping. For the RGBD classifier, the initial convolutional layer was modified to accept 4-channel inputs by concatenating the depth channel to the RGB channels, achieved by replacing the original convolutional layer with one that has 4 input dimensions. Standard pretraining on ImageNet was used helping to stabilize and accelerate training. However, for the comparison between RGB and RGBD, pretraining was not used to ensure a fair comparison, allowing both models to start from the same baseline.

4.4. Hyperparameter Optimization

Hyperparameters are the parameters that set the model, training, and optimization configurations. The set of hyperparameters can be highly influential on the performance of the model, and therefore, need to be optimized. The following section will elaborate on the use of a Differential Evolution Optimizer for hyperparameter optimization.

4.4.1. Hyperparameter Optimization Strategies

Different hyperparameter optimization techniques exist with distinct drawbacks and benefits. On one end of the spectrum there are more basic methods such as manual search, random search, and grid search. On the other end of the spectrum are more complex techniques such as gradient-based optimization, meta-heuristic approaches, and reinforcement learning-based optimization (often avoided due to their implementation complexity). These methods are typically faster than exhaustive searches but generally, similar to more basic methods, do not guarantee optimality in complex, high-dimensional problems, largely because they are probabilistic and explore the search space with limited resolution. Nonetheless, certain techniques may still prove more suitable depending on the specific problem context.

Metaheuristic approaches are problem-independent, meaning these optimization frameworks do not require specific knowledge (or gradient) of the underlying problem to explore the solution space. Several methods exist within this category, including *swarm intelligence*, *simulated annealing*, and *evolutionary techniques*. In this research, the Differential Evolution Optimizer (DEO) was chosen due to its ability to effectively control search processes while maintaining a balance between exploration and exploitation. Additionally, its implementation is intuitive, making it a practical choice for the study.

4.4.2. Differential Evolution Optimizer

Differential Evolution Optimization is a population-based method introduced in the seminal paper 'Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces' (Storn and Price, 1997). This method solves global optimization problems by iteratively improving candidate solutions through evolutionary processes. It starts with a fixed population randomly generated within the hyperparameter space. A limited set of hyperparameters was chosen to avoid an overly large search space, though determining this beforehand is challenging. In this case, the hyperparameter vectors consist of *epochs*, *learning rate*, and *batch size*. Mutation and crossover are then applied to generate new candidate solutions, represented as vectors, with the aim of finding the best solution.

The "best/1/bin" strategy is used where mutation creates a new vector by adding the weighted difference between *two* population vectors to the best vector from the current population, as can be seen in Formula 5. This introduces diversity and leverages the best solution found so far. The mutation formula for "best/1/bin" is:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}) \tag{5}$$

In equation 5, $\mathbf{v}_i$ is the mutated vector for the i^{th} individual, $\mathbf{x}_{\text{best}}$ is the best solution vector from the current population, and F is a scaling factor that weights the difference between two randomly selected vectors from the population, $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$. Crossover combines the mutant vector with the current target vector to produce a trial vector. This determines which parts of the target and mutant vectors are used, helping to refine and exploit existing solutions. The crossover can be expressed as:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if rand}_j \leq CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}, & \text{otherwise.} \end{cases} \tag{6}$$

In Equation 6, $\mathbf{u}_i$ is a trial vector, $\mathbf{v}_i$ is a mutant vector, $\mathbf{x}_i$ is the target vector, CR is the crossover rate (a constant between 0 and 1) that controls the probability of copying each component from the mutant vector, $rand_j$ is a random number between 0 and 1 for each component j, j_{rand} is a randomly chosen index to ensure at least one component from the mutant vector is used. After crossover, the trial vector is compared to the current target vector, and the one with better fitness is selected for the next generation.

To measure the fitness, a categorical cross-entropy loss is used as shown in Equation 7. In this equation, $y_i j$ is the true class label for the i-th sample, $\hat{p}_{ij}$ represents the predicted probability for the i-th sample and class j, and N is the total number of samples. The total loss is averaged over the test set, providing an overall measure of model performance for the given hyperparameter vector.

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \cdot \log(\hat{p}_{ij}) \tag{7}$$

As a stopping criterion, relative tolerance was used, defined as a fraction of the mean population objective values. The algorithm stops when the standard deviation of the population energy falls below this fraction. However, due to convergence not being achieved within the computationally acceptable limits, the process was prematurely terminated after 10 generations (660 objective evaluations). A flowchart of this algorithmic process is provided in Figure 17, and DEO parameters can be found in Table 8 (Appendix B).

To manage computational load, single-fold testing was performed using five training folds. While the optimizer, like epochs, batch size, and learning rate, is known to be an influential hyperparameter, the standard Adam optimizer was chosen for its widely recognized and reliable performance. Table 3 provides an overview of the found hyperparameters. The search space and individual objective functions are visualized in Figures 18a, 18b, and 18c (Appendix B), corresponding to obstructed holes, foreign object damage (RGB), and foreign object damage (RGBD), respectively.

Table 3
Hyperparameters *found* for Different ResNet-18 Target Models

Hyperparameters	Obstructed Holes	foreign object damage RGB	foreign object damage RGBD
Optimizer	Adam	Adam	Adam
Epochs	*37*	*49*	*45*
Learning Rate (log scale)	*-4.15*	*-4.57*	*-4.01*
Batch Size	*130*	*153*	*158*

4.5. Auxiliary Classifier Generative Adversarial Network

This section covers the key components of the AC-GAN, including *preprocessing*, *generator and discriminator architectures*, *training*, and *synthetic data generation*. The synthetic data produced by the AC-GAN will be used to augment the original dataset. Special attention is given to the adaptations necessary for applying GANs to this specific dataset. Additionally, different techniques were implemented sequentially to address instability and enhance quality.

4.5.1. Preprocessing

The data is subject to various preprocessing steps. The tiles are converted to tensor format and normalized. Horizontal and vertical flipping is used for the obstructed holes and 5% rotation is used in the foreign object damage. The preprocessing steps are summed up in the Table 9 (Appendix C).

4.5.2. Generator and Discriminator Architectures

Both the generator and discriminator in the AC-GAN required custom design due to the large-scale tiles and the intermixed feature spaces. Initial tests using a conditional GAN (cGAN) failed to produce class-specific outputs, as visually assessed, which could potentially confuse the downstream target model. Therefore, an AC-GAN was used to enforce the generation of distinct classes by leveraging the previously described additional classification loss (Section 2). As seen in Figure 12, the generator (left) and the discriminator (right) incorporate several specialized mechanisms (added sequentially) to ensure stable training.

The generator embeds the label into the label embedding space (LES). The LES and latent vector are combined, and passed to a linear layer. The linear layers is followed by five sequential blocks with deconvolutional layers, leaky Rectified Linear Units (ReLU) layers, and batch normalization layers. Finally, the generator output is constrained between -1 and 1 using a tanh activation function. The generator outputs 3-channel RGB for the obstructed holes and 4-channel RGBD in the foreign object damage.

The discriminator with either 3 or 4-channel input has six convolutional blocks. Each block consists of leaky ReLU layers, batch normalization, and dropout (dropout_rate), with dropout playing a key role in regularization for obstructed holes. A bottleneck structure is used to reduce the number of feature channels for regularization purposes. The discriminator generates two outputs: one for adversarial loss (real/fake) and another for classification loss (classes). For the obstructed holes dataset, a single-neuron with a sigmoid activation is used for the adversarial loss, paired with BCE-loss, while two-neurons with CE-loss is used for the classification loss. In the foreign object damage, two single-neurons with sigmoid activation and BCE-loss were applied, as this was empirically found to be more stable.

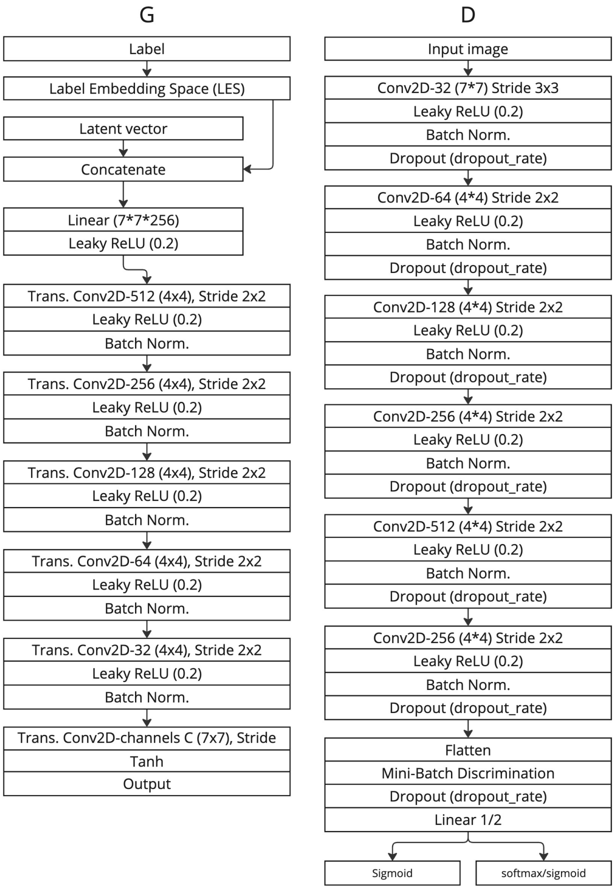


Figure 12: Generator and Discriminator architectures used in the AC-GAN. Between brackets is the kernel size.

4.5.3. Training

The AC-GAN uses an Adaptive Moment Estimation (ADAM) optimizer including three influential hyperparameters: β_1, β_2 and λ. β_1 controls the exponential decay rate for the first moment estimates (the mean of the gradients), typically set to 0.9, which helps in smoothing the gradient. β_2 controls the exponential decay rate for the second moment estimates, typically set to 0.999, which helps in smoothing the squared gradients and controlling the adaptive learning rate. Additionally, a weight_decay λ is integrated, as a regularization technique to help prevent overfitting by adding a penalty to the loss function for large weights.

As GANs are prone to mode collapse, Mini Batch Discrimination (MBD) was integrated. This mechanism evaluates the collective batch and derives a feature vector that captures not only the individual characteristics of the sample but also its relationship with the other samples in the batch. By incorporating these mini-batch statistics, MBD encourages the generator to produce a diverse set of samples.

The loss function components were weighted to give more control over the training process. The following weights were used in the AC-GAN: W_1 for discriminator adversarial

Table 4
Hyperparameter settings for obstructed holes (OH) [fold 10] and foreign object damage (FOD) [fold 1-4-6-7]

Hyperparameter	Value (OH)	Value (FOD)
channels	*3*	*4*
batch-size	*128*	*32*
latent vector	*60*	*50*
label embedding	*15*	*20*
classes	*2*	*2*
epochs	*Table 10*	*Table 10*
learning rate G	*0.0005*	*0.0005*
learning rate D	*0.0003*	*0.0005*
β_1	*0.5*	*0.5*
β_2	*0.999*	*0.999*
λ	*2e-5*	*2e-5*
step-size-scheduler	*40*	*10*
γ_d	*0.9*	*0.9*
γ_g	*0.9*	*0.9*
features MBD	*15*	*10 [20]*
ϵ_{real}	*0.85*	*0.9*
ϵ_{fake}	*0.1*	*0.1*
w1	*1*	*1*
w2	*3*	*2*
w3	*1*	*1*
w4	*0.5*	*0.5*
w5	*1*	*1*
w6	*3*	*2*
dropout	*0.45 [0.5]*	*0*

loss on real data, W_2 for discriminator classification loss on real data, W_3 for discriminator adversarial loss on fake data, W_4 for discriminator classification loss on fake data, W_5 for generator adversarial loss, and W_6 for generator classification loss. Pseudo-code A (Appendix 1) provides further insight.

Two time-scale update rule (TTUR) and stepwise rate scheduling (SRS) were used. TTUR allows the generator and discriminator to update at different learning rates, improving training stability. SRS was applied for more gradual optimization in later epochs, helping the model fine-tune as training progresses. Additionally, label smoothing for real (ϵ_{real}) and fake (ϵ_{fake}) labels was used to penalize overconfidence and smooth the backpropagation process.

4.6. Synthetic data generation

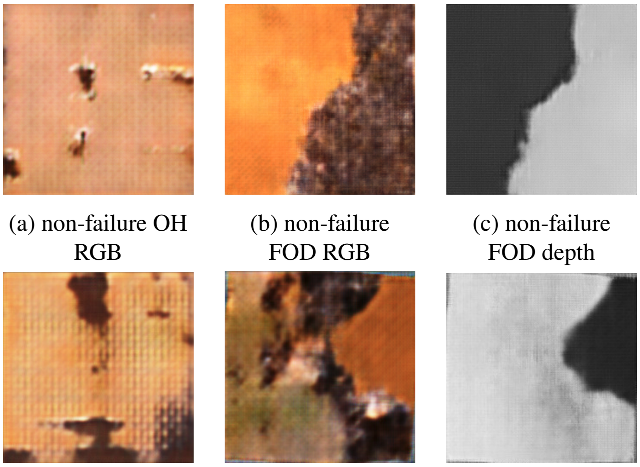
In the augmentation experiments, 10-fold cross-validation is applied to the 10 available blades to maximize data use and prevent leakage. To avoid leakage in AC-GAN augmentation, 10 separate AC-GAN models were trained and used for generation, ensuring augmented data is used only in the fold not involved in AC-GAN training. The *generator* is saved at intervals for model selection via visual assessment (Table 10, Appendix C). The generator's architecture and weights are used in combination with a latent sample from a Gaussian distribution and a class label for generation.

4.7. Visual Assessment Synthetic Data

As an important part of the research is still based on visual assessment due to the nature of the data, Figure 13 presents example outputs used in the augmentation for obstructed holes (Figures 13a and 13d) and foreign object damage (Figures 13c, 13b, 13e, 13f). These images exemplify the outputs generated by the AC-GAN used in the augmentation process. It is important to emphasize that the objective here is *not* to achieve the most photorealistic outputs, but rather to produce results with augmentative capabilities to enhance classification performance of the ResNet-18 target model.

For obstructed holes, in the non-failure class (Figure 13a), it can be visually observed that the AC-GAN emphasizes colors and textures over replicating specific details of non-failure tiles in the obstructed holes dataset. In the failure class (Figure 13d), the model gradually captures the characteristic brown spots of obstructed holes, set against a background similar in tone to the non-failure class. A general observation was that in the early stages of training, as shown in Figure 19 (Appendix C), the AC-GAN model struggles to distinguish between the non-failure and failure classes, even with the additional classification loss applied. However, as training progresses, this differentiation becomes clear and noticeable.

For foreign object damage, in the non-failure class (Figures 13b, 13c), the AC-GAN produces more abstract, darker shapes with distinct textures. In the failure class (Figures 13e, 13f), it captures the key features of foreign object damage, including the transition of the coating into the alloy at the impact location and the shiny edge typical of these areas. In Figure 20 (Appendix C), more examples can be observed including the progress over epochs.



(a) non-failure OH RGB

(b) non-failure FOD RGB

(c) non-failure FOD depth

(d) failure OH RGB

(e) failure FOD RGB

(f) failure FOD depth

Figure 13: Example synthetically generated output of the AC-GAN for both the obstructed holes (OH) (a,b) and the foreign object damage (FOD) (c,d,e,f).

5. Results and Discussion

This section presents the results and discussion of the experiments described in Methodology (Section 4). It begins with an overview of the statistical analyses used. Following this, the experimental results of RGB augmentation on the obstructed hole datasets are presented and discussed. Next, the performance comparison between RGB and RGBD data, as well as the RGBD augmentation on the foreign object dataset, is provided. In both augmentation experiments, sensitivity analysis is performed using two augmentation levels. Elaboration on the classification metrics used can be found in Supporting Work C.

5.1. Statistical Analysis

Statistical methods were used to analyze the experimental results for significance. The Shapiro-Wilk test (Shapiro and Wilk, 1965) was first applied to assess the normality of residuals in pairwise comparisons (e.g., RGB vs. RGBD) and the overall data distribution in group comparisons (e.g., different augmentation levels). This determined whether parametric or non-parametric tests were appropriate.

Based on the results of the Shapiro-Wilk and Levene's tests (Levene, 1960), different statistical tests were chosen depending on whether two sets (e.g., RGB vs. RGBD) or more sets (e.g., different augmentation levels) were analyzed. For two sets, either a *paired* t-test (Student, 1908) or Wilcoxon Signed Rank test (Wilcoxon, 1992) was used, depending on data normality, since the same data sets were compared under different treatments (augmentation).

For three or more sets, repeated measures ANOVA (Edwards, 2005) was used if the data were normally distributed; otherwise, the Friedman test (Friedman, 1937) was applied. When significant differences were found, post-hoc analyses were performed using *pairwise* t-tests or Wilcoxon tests, followed by a Bonferroni correction.

In the results, two alpha levels (α) of 0.05 and 0.1 are used to indicate the degree of statistical significance. A detailed explanation of the statistical methods and individual test results can be found in Supporting Work D.

5.2. Augmentation Performance Comparison for Obstructed Holes

The following section presents the augmentation of obstructed holes using the 3-channel AC-GAN (Section 4.5) on the 3-channel ResNet-18 target model (Section 4.3). RGB data is synthesized with augmentation applied at two levels: 50% (a balanced approach between introducing variability and maintaining the original dataset) and 200% of the original dataset size *added* (max. efficient system capacity).

Additionally, baseline performance without augmentation is included for comparison, alongside traditional on-the-fly (OTF) augmentation at 15% rotation (Section 4.3). The mean results of the 10-fold cross-validation, conducted using 10 different HPT blades, are summarized in Table 5. Figure 14 illustrates the mean values in an error bar plot, including the standard error of the mean (SEM).

In *No Augmentation* (None), the baseline performance, without any augmentation, shows an accuracy of 91.89% (Standard Deviation (SD) = 3.32), precision of 95.71% (SD = 2.04), recall of 87.74% (SD = 6.21), F1 score of 91.45% (SD = 3.72), and AUC_ROC of 97.63% (SD = 1.37).

Applying *GAN-based augmentation at 50%* (GAN50) level results in an accuracy of 93.40% (SD = 2.53), precision of 97.13% (SD = 1.47), recall of 89.46% (SD = 5.07), F1 score of 93.06% (SD = 2.86), and AUC_ROC of 98.42% (SD = 0.83). All metrics show an improvement over the baseline in means. Increased stability can be observed with reduced variability compared to baseline performance in all metrics.

Applying *GAN-based augmentation at 200%* (GAN200) level results in an accuracy increase to 94.75% (SD = 2.21), precision to 97.36% (SD = 1.82), recall to 92.03% (SD = 4.45), F1 score to 94.56% (SD = 2.39), and AUC_ROC to 98.74% (SD = 0.78). All metrics demonstrate an improved performance over baseline in means. Increased stability, similar to GAN50, can be observed with reduced variability in all metrics.

Applying *traditional augmentation* (TRADOTF) leads to improved performance across all metrics. Accuracy increases to 94.21% (SD = 3.53), precision to 97.09% (SD = 1.59), recall to 91.13% (SD = 6.36), F1 score to 93.93% (SD = 3.88), and AUC_ROC to 98.47% (SD = 1.18). Some metrics show enhanced stability, reflected by reduced variability, although this effect is not consistent across all measures.

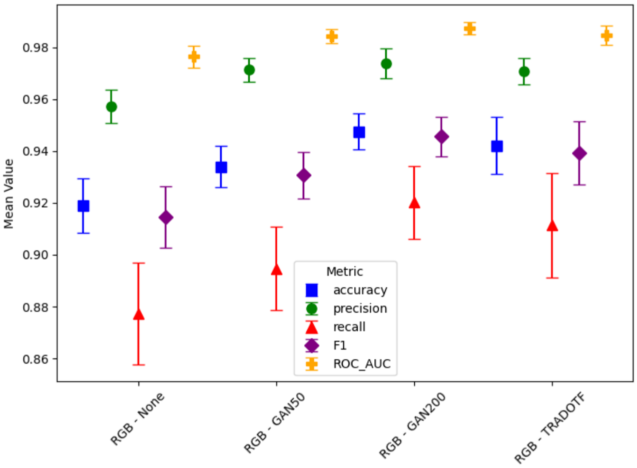


Figure 14: Error bar plot with SEM showing different standard evaluation metrics across various levels of augmentation levels in the obstructed hole dataset.

Table 5
Obstructed Holes: Comparison of augmentation techniques on classification metrics. * and ** indicate significance vs. baseline (p<0.05, p<0.1); † and †† indicate intra-group differences (p<0.05, p<0.1). Bold marks metrics above baseline, underline indicates best performance, values in parentheses are standard deviations (SD).

Augmentation Type	Level	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC_ROC (%)
No Augmentation (Baseline)	-	91.89 (3.32)	95.71 (2.04)	87.74 (6.21)	91.45 (3.72)	97.63 (1.37)
AC-GAN Augmentation	50%	**93.40*†** (2.53)	**97.13*** (1.47)	**89.46** (5.07)	**93.06*††** (2.86)	**98.42*** (0.83)
AC-GAN Augmentation	200%	**94.75*†** (2.21)	**97.36**** (1.82)	**92.03*** (4.45)	**94.56*††** (2.39)	**98.74*** (0.78)
Traditional Augmentation	TRADOTF	**94.21** (3.53)	**97.09*** (1.59)	**91.13** (6.36)	**93.93** (3.88)	**98.47*** (1.18)

Having summarized these key findings, it is important to discuss not only their implications but also the role of AC-GAN training in achieving these results. Namely, in the obstructed hole dataset, the partially overlapping feature space between the two classes was a challenge. More specifically, the general context and the distinctive failure feature—the brown obstructed spots—also appeared to some extent in the non-failure tiles. This overlap negatively impacted the discriminator's ability to distinguish between failure and non-failure cases, which in turn hindered the generator's learning process. This necessitated numerous trials to fine-tune the architectures and hyperparameters using domain-specific knowledge. This situation differs from most studies in the field, where distinctions between classes tend to be more pronounced (e.g., contrasting colors, shapes, etc.). In those cases, the discriminator can more easily establish a clear decision boundary in the feature space, enabling the generator to better model the true data distribution during the feed-forward process from the latent space.

Despite the challenges associated with GAN training, several valuable insights can be drawn from the results. As shown in Table 5, both GAN50 and GAN200 demonstrate improvements across all metrics. Notably, GAN50 shows statistically significant gains across accuracy, precision, F1 and AUC-ROC ($\alpha = 0.05$), while GAN200 achieves significant improvements in accuracy, recall, F1, AUC-ROC ($\alpha = 0.05$) and in precision ($\alpha = 0.1$). These results clearly highlight the effectiveness of GAN-based augmentation as a viable augmentation method. Moreover, the variation in augmentation levels led to statistically significant differences in both accuracy and F1 score, underscoring the model's sensitivity to augmentation volume.

When compared to the traditional augmentation method of 15% rotation (Section 4.3), the GAN-based approach delivered marginally better results. While the difference was not statistically significant, it is important to recognize that on-the-fly augmentation achieved comparative results utilizing more augmentative units (occurs during each epoch). Despite these marginal improvements, the potential gains of GAN-based augmentation should be carefully weighed against the ease and proven effectiveness of traditional augmentation techniques, which require considerably less fine-tuning, computational power and coding resources. Nonetheless, this comparison reinforces that GANs can serve as an effective tool for data augmentation, offering a comparative alternative.

The changes in performance metrics, focussing on GAN200 due to their best performance, have distinct implications for real-world applications involving HPT blades. The improved accuracy reflects the model's ability to correctly classify both failure and non-failure cases, especially given the balanced nature of the dataset. Additionally, the higher AUC-ROC indicates that the model shows improved classification performance across different decision thresholds, rather than at a specific point. However, arguably more important is the enhanced recall, achieved without (frequently encountered) loss in precision. Enhanced recall is particularly important given the high cost associated with false negatives (failing to recognize a failure) in critical HPT blades as this can have severe repercussions. While trade-offs between recall and precision are common, it is noteworthy that precision also significant improvement. This balance between recall and precision is often represented by the F1 score, the harmonic mean of the two, indicating an overall enhancement in both sensitivity and the quality of the model's predictions for failure modes. Overall, the enhanced mean performance and reduced variability suggest improved and more reliable performance if GAN-based augmentation were to be applied in in practice.

5.3. Performance Comparison: RGB vs. RGBD

This following section present the results of the performance comparison between unimodal RGB and multimodal RGBD on the 3 and 4-channel ResNet-18 target models (Section 4.3). An overview of the results is summarized in Table 6. Figure 15 displays the error bar plot with the SEM. The results are based on the mean of 10-fold cross-validation originating from 10 HPT blades using two-random seeds.

Table 6
Comparison of unimodal (RGB) and multimodal (RGBD) performance. Bold indicates improvements, with the best performance highlighted (only two). * and ** denote statistical significance at $p < 0.05$ and $p < 0.1$, respectively.

Metric	RGB Mean (SD)	RGBD Mean (SD)
Accuracy	92.88% (3.18%)	**94.13%*** (2.10%)
Precision	94.93% (2.81%)	**95.70%*** (2.59%)
Recall	90.71% (5.92%)	**92.49%**** (3.46%)
F1 Score	92.64% (3.52%)	**94.02%*** (2.16%)
ROC AUC	97.80% (1.74%)	**98.40%*** (1.23%)

The RGB input performance metrics, as shown in Table 6 and with RGB_None in Figure 15, shows an accuracy of 92.88% (SD = 3.18%), precision of 94.93% (SD = 2.81%), recall of 90.71% (SD = 5.92%), F1 score of 92.64% (SD = 3.52%), and ROC-AUC of 97.80% (SD = 1.74%).

Using RGBD input, as shown in Table 6 and with RGBD_None in Figure 15, shows that using RGBD resulted improvement across all metrics. Accuracy increased to 94.13% (SD = 2.10%), precision to 95.70% (SD = 2.59%), recall to 92.49% (SD = 3.46%), and F1 score to 94.02% (SD = 2.16%). The ROC-AUC also rose to 98.40% (SD = 1.23%).

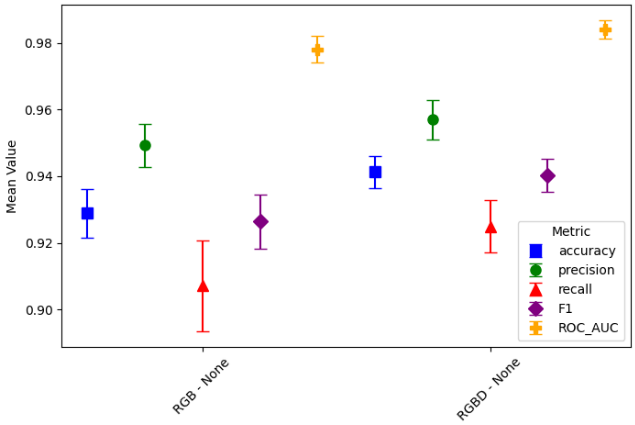


Figure 15: Error bar plot with SEM showing different evaluation metrics comparing RGB versus RGBD data inputs in the foreign object damage dataset.

It was hypothesized that using depth information from a monocular depth estimator (MiDaS) (Section 4.2) could enhance performance. The rationale was that this additional depth data would enrich feature representation, enabling the model to capture geometric properties that might be (more) discernible in depth data (than in RGB). Even though the depth estimates have limited resolution, they can provide valuable contextual insights by introducing a broader range of features that might be less noisy than the congruent RGB data. This could help reduce overfitting and improve the model's robustness to variations (e.g., lighting and texture).

Supported by statistically significant improvements in accuracy, precision, F1, and ROC-AUC ($\alpha = 0.05$) and recall ($\alpha = 0.1$), the results clearly demonstrate enhanced classification performance. Moreover, incorporating depth data not only boosts the mean performance but also reduces variability, leading to more reliable outcomes compared to using RGB data alone. These findings support further research into RGBD augmentation.

5.4. Augmentation Performance Comparison for Foreign Object Damage

This section shows the results for the foreign object damage dataset using the 4-channel ResNet-18 target model (Section 4.3). The 4-channel AC-GAN (Section 4.5) is used to synthesize RGBD data for augmentation at two levels (similar to the obstructed hole dataset): 50% and 200% of the original dataset size. Similar to the obstructed hole experiments, a baseline performance without augmentation is included for comparison, alongside traditional on-the-fly (OTF) augmentation at 15% rotation. The results are summarized in Table 7, with Figure 16 presenting the error bar plot with SEM.

In *No Augmentation* (None), baseline performance, the model achieves an accuracy of 93.58% (SD = 2.45), a precision of 96.01% (SD = 2.21), a recall of 91.00% (SD = 4.63), an F1 score of 93.36% (SD = 2.64), and an AUC_ROC of 98.26% (SD = 1.46). These values serve as the baseline performance for evaluating the impact of the augmentation techniques.

When applying *GAN-based augmentation at 50%* (GAN50), the model shows a mixed performance relative to the baseline. Accuracy drops to 91.62% (SD = 6.21), precision decreases to 92.13% (SD = 9.08), recall increases to 92.80% (SD = 4.06), the F1 score decreases to 92.05% (SD = 4.73), and AUC_ROC declines slightly to 97.87% (SD = 1.56). Besides recall, variability of the performance metrics tends to increase.

At *GAN-based augmentation at 200%* (GAN200), there is a trade-off in performance metrics compared to the baseline. Accuracy increases slightly to 93.64% (SD = 2.70), precision decreases to 94.00% (SD = 2.73), recall improves to 93.27% (SD = 3.76), F1 score slightly improves to 93.60% (SD = 2.76), and AUC_ROC is slightly lower than the baseline at 98.09% (SD = 1.49). Similar to GAN50, besides recall, variability tends to increase.

Applying *traditional augmentation* on-the-fly (OTF) results in a less pronounced trade-off in the performance metrics. Accuracy increases to 93.64% (SD = 3.95), precision decreases slightly to 95.33% (SD = 1.73), recall increases to 91.78% (SD = 7.77), F1 score remains stable at 93.36% (SD = 4.46), and AUC_ROC is slightly lower than the baseline at 98.10% (SD = 2.31). Besides precision, variability tends to increase.

Table 7
Foreign Object Damage: Comparison of augmentation techniques on classification metrics. No symbols for statistical significance are presented as there are none; bold marks metrics exceeding the baseline, underline indicates the best performance, and values in parentheses represent standard deviations (SD).

Augmentation Type	Level	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC_ROC (%)
No Augmentation (Baseline)	-	93.58 (2.45)	96.01 (2.21)	91.00 (4.63)	93.36 (2.64)	98.26 (1.46)
AC-GAN Augmentation	50%	91.62 (6.21)	92.13 (9.08)	**92.80** (4.06)	92.05 (4.73)	97.87 (1.56)
AC-GAN Augmentation	200%	**93.64** (2.70)	94.00 (2.73)	**93.27** (3.76)	**93.60** (2.76)	98.09 (1.49)
Traditional Augmentation	OTF	**93.64** (3.95)	95.33 (1.73)	**91.78** (7.77)	93.36 (4.46)	98.10 (2.31)

As can be observed in Table 7, group-based statistical tests revealed no significant differences. Consequently, post-hoc tests were not conducted, and no statistical significance was assigned.

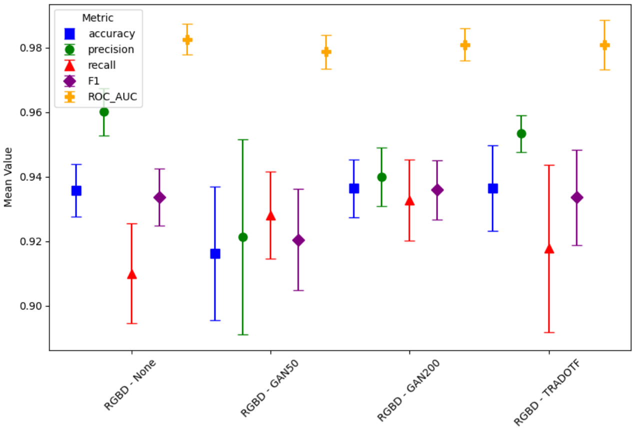


Figure 16: Error bar plot with SEM showing the different metrics across various levels of augmentation in the foreign object damage dataset.

Similar to the obstructed hole dataset, training the AC-GAN on the foreign object damage dataset posed significant challenges. The primary difficulty stemmed from the generator's tendency to converge on a limited variety of outputs, a common issue known as mode collapse. As the generator optimizes its loss, it gravitates towards the distribution where it can more easily produce realistic samples. Despite adjustments to hyperparameter settings, particularly with high MBD injection, a persistent trade-off between photorealism and diversity was observed.

Despite this training challenge, the 4-channel AC-GAN successfully generated outputs that synthesized both RGB and congruent depth, capturing the prominent features (Section 4.7). The synthesized outputs showed the model's ability to learn and represent the relationships between the blade objects and damage as foreground objects and their background.

In terms of augmentative performance, acknowledging the lack of statistical significance, several trends can be observed. GAN50 appears to lead to overall poorer performance (besides recall) and increased variability, suggesting a non-linear relationship as it might fail to reach a critical augmentation volume. However, when the augmentation is increased to GAN200, all metrics improve compared to GAN50, with more consistent performance. While some metrics remain relatively unchanged (compared to baseline), a noticeable trade-off between recall and precision appears.

When compared to the traditional augmentation method of 15% rotation, similar results appear with a slightly different degree of trade-off between recall and precision. It should be noted that this *also* did not unambiguously improve performance.

Several factors could explain these observations. First, the stable metrics suggest the model is already performing at a high level, making further improvements increasingly difficult. In terms of the trade-off, the model may be overfitting to noise or specific details in the training data rather than learning generalizable features. As a result, it becomes highly confident in its failure predictions (high precision) but less effective at identifying all failures (low recall). When exposed to GAN-based augmentation, which introduces greater variability, the model may begin to generalize more broadly. This generalization improves recall by detecting more failures but increases the risk of misclassifying non-failures, creating a trade-off between precision and recall.

Despite the lack of statistical significance, the observed trend in the trade-off between recall and precision has practical implications for real-world applications involving HPT blades. In visual inspections for foreign object damage, as in cases with obstructed holes, this trade-off may prove beneficial. The cost of false positives is often less critical than missing a failure, as engineers can perform secondary checks. In this context, the additional effort required to verify false positives is justified by the much greater risk of overlooking a potential failure.

6. Conclusion

Inherent subjectivity, inefficiencies, and the substantial cost related to human-based visual inspection of HPT blades have driven research into alternative automated techniques. The combination of DL and CV offers a compelling alternative. However, the large parameter space of more advanced DL models requires large amounts of training data, which can be both scarce and costly to obtain. This challenge has driven research into advanced augmentation techniques leveraging generative models.

This paper researched the hypothesis that GAN-based augmentation could significantly enhance failure mode classification in HPT blades. This included: a) applying GAN-based RGB augmentation for obstructed holes, b) introducing additional depth information from a monocular cross-modal depth estimator for foreign object damage, and c) building on the findings of b), introducing the approach of GAN-based RGBD augmentation for foreign object damage. The research utilized proprietary datasets from real HPT blades with synthetic failure modes applied. This led to the following key findings:

First, RGB augmentation of the obstructed holes dataset led to a statistically significant improvement in the classification performance, along with enhanced consistency. GAN-based augmentation achieved results competitive with traditional augmentation method, despite using substantially less data. This highlights the potential of GAN-based RGB augmentation for improved failure classification in HPT blade.
Second, when comparing unimodal RGB and multimodal RGBD data for foreign object damage, depth information was integrated using a monocular depth estimator through 4-channel concatenation. This led to a statistically significant improvement in classification performance and enhanced stability. This shows the potential of multimodal RGBD data for enhanced failure classification.
Lastly, RGBD augmentation was applied to the foreign object damage dataset. While this novel approach successfully generated the RGBD data, the augmentative results did not show a significant improvement, instead revealing a trade-off in performance. This made it similar to the traditional on-the-fly (OTF) augmentation method rather than offering a clear advantage. However, despite this trade-off, the approach may still hold practical value and could prove beneficial in other applications requiring multimodal data.

In conclusion, this research advanced the understanding of GAN-based augmentation and multimodal RGBD data for failure classification in HPT blade maintenance. The promising results suggest that further research could improve their integration into automated systems, offering valuable support for HPT blade inspections and guiding future advancements in the field.

7. Limitations and Future Work

This research recognizes several limitations and additional considerations, which will be described in order of the conceptual workflow of the study. Some of these limitations, together with other identified areas, provide a basis for future work.

In ML and DGMs, results are highly dependent on the specifics of the data used. In this study, balanced datasets were synthetically created using moldable material for obstructed holes and brute force for foreign object damage within a specific robotic experimental setup. While the findings are promising, caution is needed when extrapolating to non-synthetically generated datasets. Future research could extend this work by using different robotic configurations and unbalanced datasets of real failure modes. Additionally, exploring various failure modes with different data volumes and qualities could provide a better understanding of the applicability of GAN models in real-world contexts.

MiDaS was used as a monocular depth estimator to construct the congruent depth maps of the RGB imagery. Despite the promising results, it is recognized that the depth maps generated by MiDaS have limitations in quality (which can be beneficial given the simplicity). Future work could focus on using different cross-modal models or construct relevant (simulated) RGBD data to further *fine-tune* the (open source) MiDaS model, enhancing its performance on datasets specific to the research

While GANs have a unique working mechanism that theoretically enables them to be effective even with limited data, they are notoriously unstable and thus particularly challenging to train. It's important to acknowledge that other DGMs exist with the ability to augment which might have different, more stable, training properties given the dataset (difficult to determine a priori). Other DGMs, such as Variational Auto-encoders (VAEs) and Diffusion Models have advanced notably in the past years. Although these models may be theoretically more susceptible to low data volumes, future research could explore their potential.

One of the more significant limitations and challenges in DGMs, particularly in GANs, is configuring hyperparameters, including the network architecture. This configuration directly influences the quality of the synthesized data used for augmentation. GANs, as noted, are particularly difficult to train due to their adversarial mini-max construct. This challenge is further compounded by the low data volumes. Tuning these models is often still performed manually, guided by general heuristics, making it a *time-consuming* and *protracted* process. Future research could focus on developing and testing robust hyperparameter optimization strategies specifically for GANs.

Regarding the developed model, although a broad set of performance-enhancing methods were employed, additional techniques like alternative loss functions (e.g., Wasserstein Loss), Spectral Normalization, or progressive growing (particularly valuable given the high resolution) could further boost performance. Furthermore, future research could refine this model by experimenting with variations in architecture.

The results of the RGB augmentation on the obstructed hole dataset clearly demonstrated strong potential. However, the intermediate outcomes from the foreign object damage dataset in the RGBD domain should not be overlooked. The improved performance from incorporating depth information, compared to using RGB alone, combined with the ability to effectively generate RGBD data, opens valuable opportunities for future research. Despite the absence of a clear advantage (namely a trade-off) or statistical significance in this specific dataset, the combination of RGBD data and GAN-based techniques suggests a promising avenue for further exploration and refinement in future studies.

Finally, an interesting direction for future research is to explore feature fusion GAN models as an alternative to traditional pixel-level GAN models. Instead of approximating the distribution of the original data at the pixel level, this approach would focus on reconstructing an intermediate network layer where features have already been extracted. This method could be particularly valuable not only for RGB models but also for RGBD, where the features of both RGB and depth are already fused, potentially allowing for more efficient and effective augmentation.

Acknowledgements

To start with, I would like to express my gratitude to the Royal Netherlands Aerospace Center for enabling this research and providing me with the invaluable opportunity to pursue a field that deeply interests me. Additionally, I thank the Technical University of Delft, particularly the Department of Control and Operations within Aerospace Engineering, for providing me with great education over the past few years.

I would also like to express my deep gratitude to several individuals who have been instrumental in this research. I extend my thanks to Marcia Baptista and Madelyn Deuten. Your expertise in the field, continuous push and your guidance in developing critical thinking skills have been immensely valuable. In addition, I'm greatfull and appreciative of the flexibility and consideration shown during my concussion. I would like to thank the board of examiners and, especially Ingeborg de Pater, for bringing the research to completion.

Finally, I would like to thank my father, Prof. Dr. Jan IJzermans, along with Jasper van Leuven, Storm Holman, and Mathieu Valkier for the numerous helpful discussions.

Nomenclature

AC-GAN Auxiliary Classifier GAN

AET Acoustic Engine Testing

ARM Auto Regressive Model

BCE-loss Binary-Cross Entropy loss

CE-loss Cross Entropy loss

CNN Convolutional Neural Network

CV Computer Vision

cGAN Conditional Generative Adversarial Network

D Discriminator

DEO Differential Evolution Optimizer

DGM Deep Generative Model

DL Deep Learning

DNN Deep Neural Network

EBM Energy Based Model

FOD Foreign Object Damage

GAN Generative Adversarial Network

GenX General Electric Next Generation

G Generator

HPT High Pressure Turbine

ITT Infrared Thermography Testing

KLM Royal Dutch Airline

LVM Latent Variable Model

MPT Magnetic Particle Testing

NLR Royal Netherlands Aerospace Centre

OH Obstructed Holes

PD Preventative Maintenance

PdM Predictive Maintenance

RGB Red Green Blue

RGBD Red Green Blue Depth

RM Reactive Maintenance

VAE Variational Autoencoder

AUC Area Under the Curve - Receiver Operating Characteristic

HPT High-Pressure Turbine

ML Machine Learning

CV Computer Vision

OTF On The Fly

SEM Standard Error of the Mean

ResNet Residual Network

OH Obstructed Hole

FOD Foreign Object Damage

TTUR Two Timescale Update Rule

ReLU Rectified Linear Unit

LES Label Embedding Space

DC Deep Convolutional

Grad-CAM Gradient-weighted Class Activation Mapping

Formulas

$P(x)$ Probability distribution of data x

$P(z)$ Probability distribution of latent variable z, typically a standard Gaussian

x_i Individual data point in a sequence

$\mathbb{R}^d \to \mathbb{R}^d$ Space to space mapping

θ_D Parameters of the Discriminator

θ_G Parameters of the Generator

N Batch size

$x^{(i)}$ Real data samples

$z^{(i)}$ Noise samples

x' Generative domain

$\mathbb{E}$ Expected value

$\mathcal{L}$ Loss function

$P(y|x)$ Conditional probability of y given x

$G(z)$

References

Abbas, A., Jain, S., Gour, M., Vankudothu, S., 2021. Tomato plant disease detection using transfer learning with c-gan synthetic images. Computers and Electronics in Agriculture 187, 106279.

Ahmad, Z., Jaffri, Z.u.A., Chen, M., Bao, S., 2024. Understanding gans: fundamentals, variants, training challenges, applications, and open problems. Multimedia Tools and Applications , 1–77.

Antoniou, A., Storkey, A., Edwards, H., 2017. Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 .

Aust, J., Pons, D., 2019. Taxonomy of gas turbine blade defects. Aerospace 6, 58.

Aust, J., Pons, D., 2022. Comparative analysis of human operators and advanced technologies in the visual inspection of aero engine blades. Applied Sciences 12, 2250.

Aust, J., Shankland, S., Pons, D., Mukundan, R., Mitrovic, A., . Automated defect detection and decision-support in gas turbine blade inspection. Aerospace 8, 30.

Aust, J., Shankland, S., Pons, D., Mukundan, R., Mitrovic, A., 2021. Automated defect detection and decision-support in gas turbine blade inspection. Aerospace 8, 30.

Birkl, R., Wofk, D., Müller, M., 2023a. Midas v3. 1–a model zoo for robust monocular relative depth estimation. arXiv preprint arXiv:2307.14460 .

Birkl, R., Wofk, D., Müller, M., 2023b. Midas v3. 1–a model zoo for robust monocular relative depth estimation. arXiv preprint arXiv:2307.14460 .

Edwards, A.W., 2005. Ra fischer, statistical methods for research workers, (1925), in: Landmark writings in western mathematics 1640-1940. Elsevier, pp. 856–870.

Franz, S., Rottoli, M., Bertram, C., 2022. The wide range of possible aviation demand futures after the covid-19 pandemic. Environmental Research Letters 17, 064009.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H., 2018. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. Neurocomputing 321, 321–331.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association 32, 675–701.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems 27.

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.

He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q., 2022. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. arXiv preprint arXiv:2211.13221 .

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al., 2022. Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 .

Hu, G., Wu, H., Zhang, Y., Wan, M., 2019. A low shot learning method for tea leaf's disease identification. Computers and Electronics in Agriculture 163, 104852.

Hussain, Z., Gimenez, F., Yi, D., Rubin, D., 2017. Differential data augmentation techniques for medical imaging classification tasks, in: AMIA annual symposium proceedings, American Medical Informatics Association. p. 979.

Juarez, R., Gutierrez, N., Petersen, E.L., 2023. High-temperature degradation and coking of aircraft gas turbine engine lubricants, in: AIAA SCITECH 2023 Forum, p. 1252.

Karras, T., Aila, T., Laine, S., Lehtinen, J., 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 .

Kauffman, R.E., Feng, A., Karasek, K.R., 2000. Coke formation from aircraft turbine engine oils: Part i—deposit analysis and development of laboratory oil coking test. Tribology transactions 43, 823–829.

Khosla, C., Saini, B.S., 2020. Enhancing performance of deep learning models with different data augmentation techniques: A survey, in: 2020

International Conference on Intelligent Engineering and Management (ICIEM), IEEE. pp. 79–85.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25.

Levene, H., 1960. Robust tests for equality of variances. Contributions to probability and statistics , 278–292.

Li, B., Qi, X., Lukasiewicz, T., Torr, P., 2019a. Controllable text-to-image generation. Advances in Neural Information Processing Systems 32.

Li, Y., Tang, S., Zhang, R., Zhang, Y., Li, J., Yan, S., 2019b. Asymmetric gan for unpaired image-to-image translation. IEEE Transactions on Image Processing 28, 5881–5896.

Lindsay, G.W., 2021. Convolutional neural networks as a model of the visual system: Past, present, and future. Journal of cognitive neuroscience 33, 2017–2031.

Liu, Z.S., Siu, W.C., Chan, Y.L., 2020. Photo-realistic image super-resolution via variational autoencoders. IEEE Transactions on Circuits and Systems for video Technology 31, 1351–1365.

Madani, A., Moradi, M., Karargyris, A., Syeda-Mahmood, T., 2018a. Chest x-ray generation and data augmentation for cardiovascular abnormality classification, in: Medical imaging 2018: Image processing, SPIE. pp. 415–420.

Madani, A., Moradi, M., Karargyris, A., Syeda-Mahmood, T., 2018b. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation, in: 2018 IEEE 15th International symposium on biomedical imaging (ISBI 2018), IEEE. pp. 1038–1042.

Maddox, M., . Human Factors Guide for Aviation Maintenance and Inspection. https://www.faa.gov/sites/faa.gov/files/about/initiatives/maintenance_hf/training_tools/HF_Guide.pdf. [Accessed 19-08-2024].

Marais, K.B., Robichaud, M.R., 2012. Analysis of trends in aviation maintenance risk: An empirical approach. Reliability Engineering & System Safety 106, 104–118.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 .

Motamed, S., Rogalla, P., Khalvati, F., 2021. Data augmentation using generative adversarial networks (gans) for gan-based detection of pneumonia and covid-19 in chest x-ray images. Informatics in Medicine Unlocked 27, 100779.

Odena, A., Olah, C., Shlens, J., 2017. Conditional image synthesis with auxiliary classifier gans, in: International conference on machine learning, PMLR. pp. 2642–2651.

Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 .

Pandey, S., Singh, P.R., Tian, J., 2020. An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation. Biomedical Signal Processing and Control 57, 101782.

Prince, S.J., 2023. Understanding Deep Learning. MIT Press. URL: http://udlbook.com.

Ranftl, R., Bochkovskiy, A., Koltun, V., 2021. Vision transformers for dense prediction. ICCV .

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V., 2022. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence 44.

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52, 591–611.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11, 341–359.

Student, 1908. The probable error of a mean. Biometrika , 1–25.

Sun, X., Wandelt, S., Zhang, A., 2023. A data-driven analysis of the aviation recovery from the covid-19 pandemic. Journal of Air Transport Management 109, 102401.

Tan, Y., Li, Y., Liu, H., Lu, W., Xiao, X., 2020. Performance comparison of data classification based on modern convolutional neural network architectures, in: 2020 39th Chinese Control Conference (CCC), IEEE. pp. 815–818.

Taylor, L., Nitschke, G., 2018. Improving deep learning with generic data augmentation, in: 2018 IEEE symposium series on computational intelligence (SSCI), IEEE. pp. 1542–1547.

Tomczak, J.M., 2022. Deep Generative modeling. Springer.

Uludag, A., 2016. The magnetic particle inspection examination of aircraft propeller mounting bolts. Journal of Multidisciplinary Engineering Science and Technology 3, 1–5.

Wilcoxon, F., 1992. Individual comparisons by ranking methods, in: Breakthroughs in statistics: Methodology and distribution. Springer, pp. 196–202.

Wu, N., Zong, Z.M., Fei, Y.W., Ma, J., 2017. Studies on thermal oxidation stability of aviation lubricating oils, in: MATEC Web of Conferences, EDP Sciences. p. 02002.

Zhan, X., Han, S., Rong, N., Cao, Y., 2023. A hybrid transfer learning method for transient stability prediction considering sample imbalance. Applied Energy 333, 120573.

Zhang, Z., Yang, G., Hu, K., 2018. Prediction of fatigue crack growth in gas turbine engine blades using acoustic emission. Sensors 18, 1321.

A. Pseudo-code Auxiliary Classifier Generative Adversarial Network

Algorithm 1 offers a simplified overview of the core learning process in an AC-GAN. Unlike a standard GAN, this code incorporates a class-based differentiation loss within the discriminator, which plays a crucial role in updating both the discriminator and the generator.

Algorithm 1 Minibatch stochastic gradient descent training of Auxiliary Classifier GANs

```
for number of training epochs do
    Sample minibatch of $m$ real examples $\{(x^{(1)}, c^{(1)}), \dots, (x^{(m)}, c^{(m)})\}$ from the data distribution $p_{data}(x, c)$
    Sample minibatch of $m$ noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$
    Generate minibatch of $m$ fake examples $\{G(z^{(1)}, c^{(1)}), \dots, G(z^{(m)}, c^{(m)})\}$ using the Generator
    Update the Discriminator:
        Compute Discriminator real adversarial loss: $disc\_adversarial\_real\_loss$
        Compute classification loss for real examples: $disc\_classification\_real\_loss$
        Compute total real loss: $d\_loss\_real = w1 \times disc\_adversarial\_real\_loss + w2 \times disc\_classification\_real\_loss$
        Compute Discriminator fake adversarial loss: $disc\_adversarial\_fake\_loss$
        Compute classification loss for fake examples: $disc\_classification\_fake\_loss$
        Compute total fake loss: $d\_loss\_fake = w3 \times disc\_adversarial\_fake\_loss + w4 \times disc\_class.\_fake\_loss$
        Total Discriminator loss: $d\_loss\_total = d\_loss\_real + d\_loss\_fake$
        Update the Discriminator by backpropagating $d\_loss\_total$
    Update the Generator:
        Compute Generator adversarial loss: $g\_adversarial\_loss$
        Compute classification loss: $g\_classification\_loss$
        Total Generator loss: $g\_loss\_total = w5 \times g\_adversarial\_loss + w6 \times g\_classification\_loss$
        Update the Generator by backpropagating $g\_loss\_total$
end for
```

B. Additional Material Differential Evolution Optimizer

Figure 17 presents a flowchart that outlines the algorithm underlying the DEO, including the steps of initialization, mutation, crossover, and selection of the best fit. The optimizer terminates when a stopping condition is met, which occurs either when the maximum number of objective evaluations is reached or when the standard deviation of the population (a measure of spread) falls below a relative tolerance, defined as a fraction of the absolute value of the population mean. It's important to note that the term 'optimal solution' here refers to the algorithm approaching the optimal solution based on the stopping condition, rather than guaranteeing its exact attainment.

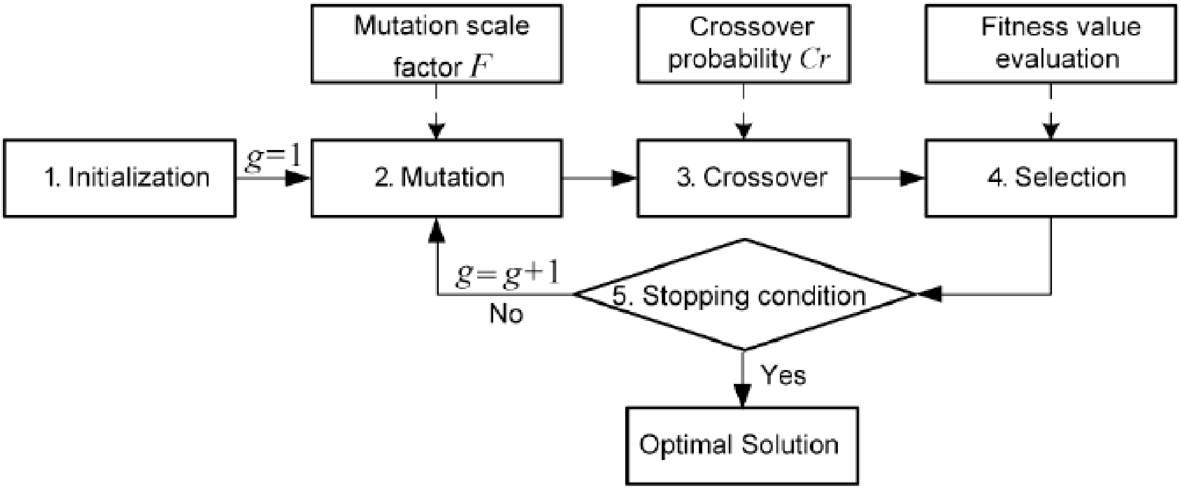


Figure 17: Flowchart of the Differential Evolution (DE) algorithm, illustrating the key steps: Initialization, Mutation, Crossover, Selection, and the evaluation of the stopping condition.

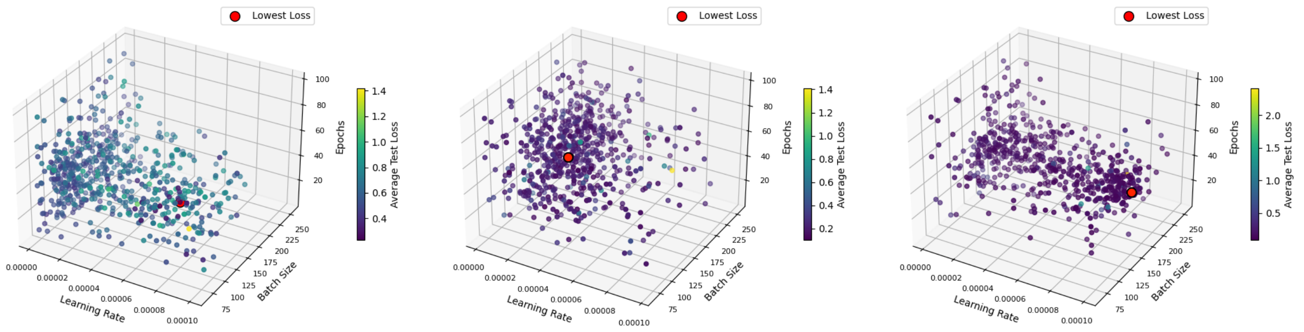
Table 8 lists the DEO parameters: Best1bin strategy, empirically set population size and iterations, balanced tolerance, mutation, and recombination, with Polish step disabled due to computational constraints.

Table 8

Configuration Parameters for the Differential Evolution Optimization Process

Parameter	Value
Strategy	best1bin
Max Iterations	10
Population Size	15
Mutation Range	(0.3, 0.6)
Recombination	0.6
Tolerance	0.1
Display	True
Polish	False
Callback	evolution_callback

Figure 18 illustrates the evolutionary process within the search space across the dimensions of epoch, batch size, and learning rate, highlighting the balance between exploration and exploitation throughout the optimization. The best solution vector is depicted with a red circle.



(a)

(b)

(c)

Figure 18: Visual overview of the hyperparameter optimization algorithm applied to the obstructed holes dataset (a), the foreign object damage dataset using RGB (b), and the foreign object damage dataset using RGBD (c).

C. Additional Material Auxiliary Classifier Generative Adversarial Network

Table 9 represent the preprocessing steps used in the AC-GAN.

Table 9

Image Preprocessing and Augmentation Techniques

Transformation	**Description**
Random Horizontal Flip (only obstructed hole)	Randomly flip the image horizontally with a probability of 0.5.
Random Vertical Flip (only obstructed hole)	Randomly flip the image vertically with a probability of 0.5.
Random Rotation (only foreign object damage)	Rotate the image by up to 5 degrees.
Resize (redundanct as already 224 by 224)	Resize the image to 224x224 pixels.
ToTensor	Convert the image to a tensor.
Normalize RGB channels	Normalize with mean = [0.5, 0.5, 0.5] and standard deviation = [0.5, 0.5, 0.5].
Normalize depth channel (only foreign object damage)	Normalize with mean = [0.5] and standard deviation = [0.5].

Table 10 shows the exact epochs at which the generator weights are taken to be used for data synthesis.

Table 10
Epochs were manually selected based on visual inspection, identifying the point where the generated output best resembles the original data.

Fold	1	2	3	4	5	6	7	8	9	10
obstructed hole	84	84	90	84	108	102	114	84	78	114
foreign object damage	39	72	33	51	57	72	63	27	51	66

Figure 19 illustrates examples from the training process of the obstructed hole dataset, showcasing the top five non-failure and bottom five failure cases, to provide insight into the outputs generated by the AC-GAN. Note, the differences are nuanced and can be seen upon close examination.

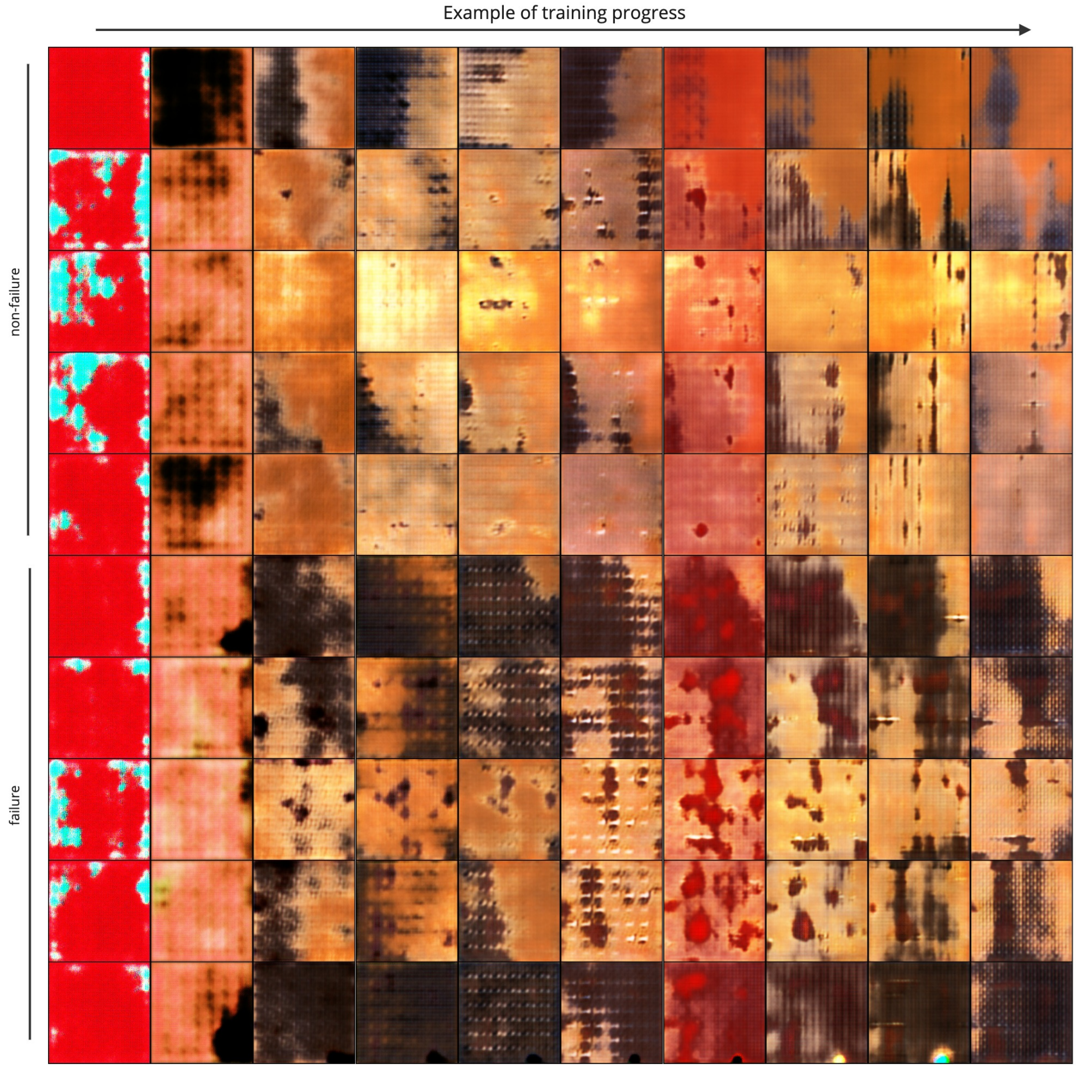


Figure 19: Example of the training progress in the obstructed holes. It can be observed that the brown characteristic spots, though subtle, are visible in the failure set (and not in the non-failure set) upon closer inspection.

Figure 20 presents examples from the training process on the foreign object damage dataset. Rows 1-3 display non-failure RGB cases, rows 4-6 show failure RGB cases, rows 7-9 depict non-failure depth cases, and rows 10-12 illustrate failure depth cases. These examples offer insight into the outputs generated by the AC-GAN.

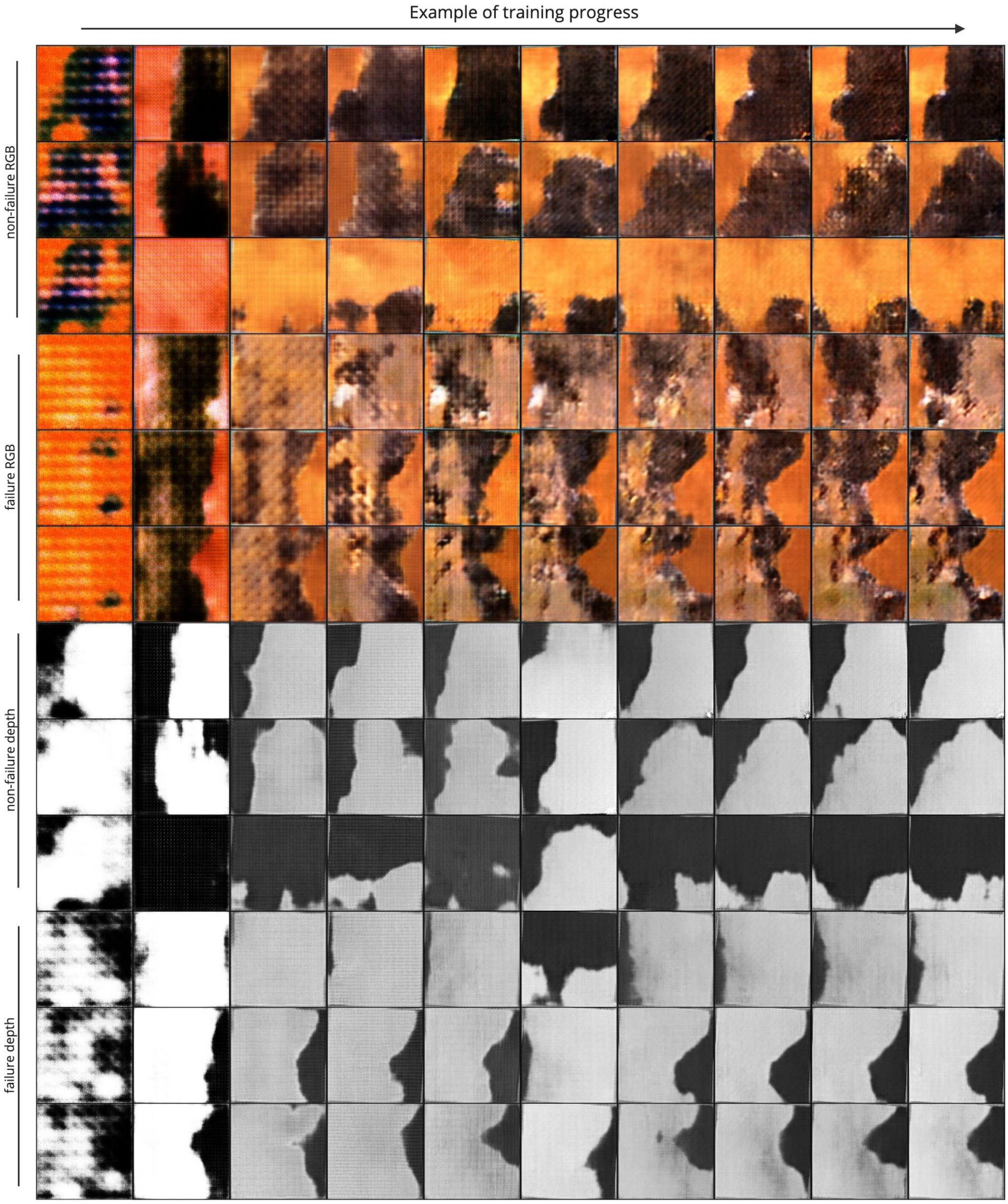


Figure 20: Training progress examples for the foreign object damage dataset reveal distinct patterns: the non-failure class converges into more uniform black shapes, while the failure classes progressively become more distinct and recognizable. Over time, the coating transitions to a damaged alloy, with light reflecting on the damaged alloy.