A Hybrid Game Engine–Generative AI Framework for Overcoming Data Scarcity in Open-Pit Crack Detection

Rohan Le Roux *, Siavash Khaksar, Mohammadali Sepehri, Iain Murray

Posted Date: 12 March 2026

doi: 10.20944/preprints202603.0954.v1

Keywords: synthetic dataset generation; generative adversarial networks; computer vision; deep learning; object detection; surface crack detection; open-pit mining

 doi:10.20944/preprints202603.0954.v1

Article

A Hybrid Game Engine—Generative AI Framework for Overcoming Data Scarcity in Open-Pit Crack Detection

Rohan Le Roux [1,*], Siavash Khaksar [1], Mohammadali Sepehri [2] and Iain Murray [2]

[1] School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Perth 6102, Australia

[2] The Western Australian School of Mines, Curtin University, Kalgoorlie 6430, Australia

* Correspondence: rohan.leroux@curtin.edu.au

Abstract

Open-pit mining relies heavily on visual inspection to identify indicators of slope instability such as surface cracks. Early identification of these geotechnical hazards allows for the implementation of safety interventions to protect both workers and assets in the event of slope failures or landslides. While computer vision (CV) approaches offer a promising avenue for autonomous crack detection, their effectiveness remains constrained by the scarcity of labelled geotechnical datasets. Deep learning (DL) models require large amounts of representative training data to generalize to unseen conditions; however, collecting such data from operational mine sites is limited by safety, cost, and data confidentiality constraints. To address this challenge, we propose a hybrid game engine—generative artificial intelligence (AI) framework for large-scale dataset synthesis. Leveraging a parameterized virtual environment developed in Unreal Engine 5 (UE5), the framework captures realistic images of open-pit surface cracks and enriches their visual diversity using StyleGAN2-ADA. The resulting datasets were used to train the YOLOv11 real-time object detection model and evaluated on a real-world dataset of open-pit slope imagery to assess the effectiveness of the proposed framework in improving CV model generalizability under extreme data scarcity. Experimental results demonstrated that models trained on the proposed framework substantially outperformed the UE5 baseline, with average precision (AP) at intersection over union (IoU) thresholds of 0.5 and [0.5:0.95] increasing from 0.403 to 0.922 and 0.223 to 0.722 respectively, accompanied by a reduction in missed detections from 95 to eight for the best-performing configurations. These findings demonstrate the potential of hybrid generative AI frameworks to mitigate data scarcity in CV applications and support the development of scalable automated slope monitoring systems for improved worker safety and operational efficiency in open-pit mining.

Keywords: synthetic dataset generation; generative adversarial networks; computer vision; deep learning; object detection; surface crack detection; open-pit mining

1. Introduction

Open-pit mines are among the most hazardous industrial environments due to the risks posed by geotechnical events such as slope failures and landslides [1–3]. The devastating consequences of these incidents are exemplified by recent disasters such as the 2023 Xinjing coal mine landslide in China, which resulted in 53 fatalities and approximately USD 28 million in economic losses [4], and the 2020 Hpakant jade mine landslide in Myanmar, which claimed nearly 200 lives and severely impacted local communities [5]. These events can be triggered by several factors, including weak geological structures [6], intense or prolonged rainfall [7,8], seismic activity, and vibrations from excavation and blasting [9]. Early warning systems are therefore designed to monitor slope displacement and detect hazards such as surface cracks [10–12], enabling safety interventions such

 doi:10.20944/preprints202603.0954.v1

as exclusion zones to protect both workers and assets [13]. Despite advances in technologies such as stability radar, visual inspection remains central to hazard identification across many open-pit mines in Australia [14–16]. However, this manual practice is both labor-intensive and subjective, exposing workers to hazardous environments and compromising safety [17–19].

Consequently, recent studies have focused on developing automated approaches that leverage technologies such as AI to reduce dependence on manual monitoring while enhancing operational safety and efficiency [20]. Specifically, advances in DL [21] have driven substantial progress in CV tasks such as object detection, image classification, and image segmentation [22], thereby enabling machines to derive meaningful information from real-world visual data [23]. To that end, neural network architectures such as convolutional neural networks (CNNs) [24] and vision transformers (ViTs) [25] have been widely utilized across diverse domains [26–35], underscoring their potential for geotechnical risk management. In particular, recent studies have demonstrated the use of CNN-based CV models such as YOLOv8 [36], YOLOv10 [37], Mask R-CNN [38], U-Net [39,40], and ENet [41] for automated surface crack detection in open-pit mining. While these works demonstrate the feasibility of DL models for geotechnical hazard identification, their effectiveness remains constrained by the limited availability of labelled crack images.

This issue, commonly referred to as data scarcity, is a ubiquitous problem in the field of DL [42], where domain generalization, or the capacity of a CV model to recognize objects in unseen settings or environments [43], is driven by the volume and representativeness of the data used for training [44]. Data scarcity is especially pronounced in industrial domains such as mining [45], where datasets are inherently commercially sensitive and limited by the significant cost and expertise required for data collection and annotation [46–48]. To address this challenge, techniques such as transfer learning [49] and data augmentation [50] have been widely adopted in the literature, particularly in healthcare and other data-constrained domains. Transfer learning reduces reliance on large datasets but is vulnerable to source-target domain mismatch [51,52], while data augmentation, though capable of artificially increasing dataset size [53], is prone to amplifying existing distributional biases [54,55]. As both methods remain fundamentally bounded by the quality of the original training data, interest continues to grow in techniques capable of generating entirely new and diverse training data at scale [56].

One established approach is the use of game engines, where platforms such as Unreal Engine (UE) [57] and Unity [58] are adapted to render synthetic images for training CV models. These programs enable controlled variation of scene parameters and environmental conditions, as well as automated dataset generation and ground-truth annotation [59]. However, their diversity is inherently bounded by the manual effort required for scene and asset development, imposing practical limitations on dataset scale and variation. Generative models address this limitation by learning directly from existing real-world distributions to produce high-fidelity outputs at scale. Among these, generative adversarial networks (GANs) [60] such as StyleGAN2-ADA are highly effective at emulating realistic visual patterns [61], providing controllable latent-space manipulation and adaptive augmentation that mitigates discriminator overfitting for small datasets [62]. Although modern diffusion models can also produce visually rich and highly stylized imagery, their outputs are more dependent on prompt tuning and exhibit less fine-grained, deterministic control compared to GANs, making them less suited to domains requiring strict structural realism and reproducibility [63]. Despite these technological advances, the generation of synthetic images remains constrained by a number of limitations. The most impactful of these issues is the reality gap, a concept which refers to the perceptual and distributional disparity between real and synthetic data, a factor which often limits the generalizability of CV models trained on images from game engines [64]. Moreover, generative models typically require substantial training data, impacting their effectiveness in domains where real-world datasets are limited [65]. While techniques such as domain randomization [66,67] have been explored as a means of mitigating the reality gap [68], no prior study has systematically evaluated if game engines can be leveraged to train generative models for realistic and scalable image synthesis as a mitigant for data scarcity.

 doi:10.20944/preprints202603.0954.v1

To address this gap, we develop and evaluate a hybrid game engine—generative AI framework that produces synthetic images to offset data scarcity and enhance the generalizability of CV models. The main contributions of this work are summarized as follows:

- We develop a scalable hybrid framework integrating UE5 and StyleGAN2-ADA to generate realistic synthetic images with automated annotations for training CV models.
- We demonstrate that synthetic images of surface cracks generated through our pipeline achieve enhanced fidelity and diversity compared to game engine data alone, validated quantitatively through Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS).
- We evaluate the downstream effectiveness of images generated through our pipeline by training the real-time object detection model YOLOv11, achieving substantial performance improvements relative to models trained solely on game engine data.

To the best of our knowledge, this study presents the first systematic evaluation of a hybrid game engine—generative AI framework for data synthesis utilizing UE5 and StyleGAN2-ADA. By combining the geometric and structural realism of game engines with the scalable diversity and textural domain adaptation of generative modelling, our framework demonstrates improved generalization performance in data-scarce CV tasks. Applied to surface crack detection in open-pit mining, the framework enhances the accuracy of object detection models in identifying slope failure precursors, supporting improved safety outcomes while demonstrating a methodology with potential transferability to other data-constrained domains.

2. Related Works

2.1. Synthetic Dataset Generation Using Game Engines

Advances in game engine technology have pushed the boundaries of visual realism and provided a promising means of addressing data scarcity in CV. Unlike manual data collection, which can be costly and time-consuming, game engines enable synthetic data generation with automated ground-truth annotation and fine-grained control of scene parameters and environmental conditions [69]. For instance, Half-Life 2's Source engine [70], renowned for its detailed and lifelike animation and physics technology, was first utilized nearly two decades ago to develop and test an autonomous surveillance system [71]. Since then, researchers have increasingly leveraged synthetic data generated by photorealistic game engines to train and validate CV models for tasks such as object detection and image segmentation [72]. Table 1 summarizes recent works which apply game engine synthetic data across diverse applications, from construction safety monitoring to autonomous vehicle detection. These studies demonstrate performance improvements ranging from modest gains of less than 1 % [73] to substantial enhancements exceeding 65 % [74], highlighting the variable influence of game engine data on model generalizability. The most common development platforms used are UE4 and Unity due to their accessibility and integration with CV plugins such as NDDS [75] and Unity Perception Package [76], which support domain randomization techniques and automatically generate ground-truth labels such as bounding boxes and segmentation masks for captured data. An alternative approach is the use of commercial games, such as Grand Theft Auto V (GTA V) [77], where a combination of in-engine tools and community mods are adapted for synthetic data capture and annotation [78]. Collectively, these works demonstrate the versatility of game engines for synthetic data generation, producing datasets ranging from approximately 1 500 [79] to over one million images [78]. Studies utilizing fewer than 3 000 synthesized images for training of downstream CV models reported limited performance improvements or underfitting [73,79], suggesting insufficient data diversity to support robust model generalization. In contrast, approaches which synthesized larger training datasets demonstrated substantial gains [74,80,81], highlighting the impact of dataset size on downstream performance.

 doi:10.20944/preprints202603.0954.v1

Table 1. Summary of related works on synthetic dataset generation using game engines.

Application	Downstream Task	Platform	Dataset Size	Performance
Construction monitoring [74]	Object detection	Unity	7 000	mAP@[0.5:0.95]: 0.46
Generic object detection [79]	Object detection	UE4 with NDDS	1 500	Not reported
Autonomous driving [80]	Object detection	UE5	16 700	mAP@0.5: 0.67
Navigation assistance [73]	Object detection	UE4 with NDDS	3 000	Precision: 0.92 Recall: 0.91
Exercise monitoring [82]	Pose estimation	Unity	5 000	I3D test accuracy: 0.99
Grocery item detection [81]	Object detection	Unity	400 000	mAP@[0.5:0.95]: 0.68
Warehouse object detection [83]	Semantic segmentation	Unity	7 140	mAP@0.5: 0.65
Autonomous driving [78]	Semantic segmentation	GTA V	1 355 568	CIoU: 0.45
Animal monitoring [84]	Pose estimation	Unity	32 000	PCK: 0.13
Construction monitoring [85]	Object detection	Unity	6 000	Precision: 0.92
Generic object detection [86]	Classification	UE4	31 200	Top-1 accuracy: 0.72

Abbreviations: *mAP* – Mean AP; *NDDS* – NVIDIA DL Dataset Synthesizer; *I3D* – Inflated 3D Networks; *CIoU* – Complete Intersection over Union; *PCK* – Percentage of Correct Keypoints.

These findings indicate that while game engines enable scalable data generation, meaningful performance benefits require datasets of sufficient scale and diversity to capture the contextual variability necessary for effective transferability.

In addition to dataset size, the realism of rendered images remains a key determinant of downstream performance. Several studies have demonstrated that models trained on data from high-fidelity game engines tend to generalize better than those trained on less photorealistic platforms [83]. For instance, [80] reported that the enhanced texture detail, lighting, and reflections of images generated in UE5 led to a 17.3 % increase in mAP relative to a model trained on data produced in UE4, highlighting the influence of visual fidelity on learned feature distributions for CV models. Similarly, commercial games, developed with significant budgets and advanced rendering pipelines, typically produce more realistic images than those generated by less robust engines such as Unity (see Figure 1) [78]. As a result, models trained on lower fidelity datasets often struggle to generalize in real-world applications because their limited realism widens the reality gap [83].

 doi:10.20944/preprints202603.0954.v1

(**a**) (**b**)

Figure 1. Graphical comparison between (**a**) GTA V and (**b**) a nature scene in Unity. GTA V demonstrates greater photorealism than Unity due to advanced rendering effects such as ray-traced reflections, global illumination, and detailed material texturing, illustrating the disparity in visual fidelity that contributes to the reality gap.

To address this issue, several studies employ transfer learning by pre-training using synthetic data and fine-tuning with small quantities of real-world data, thereby better aligning the synthetic and real feature domains. This mixed dataset approach, demonstrated in [84], resulted in a 5 % improvement in the PCK for pose estimation relative to a model trained solely on synthetic data. Furthermore, [81] reported a 79.5 % improvement in mAP when fine-tuning a model trained on 400 000 synthetic images with only 760 real images (a 526:1 synthetic-to-real ratio), underscoring the effectiveness of limited real-world data in enhancing model generalizability. Complementary to this, domain randomization techniques have proven effective in bolstering model robustness across unseen conditions. For example, [85] demonstrated a 22.8 % increase in mAP through successive randomization of parameters such camera positioning and object location, while [73,79] and [81] leveraged NDDS and Unity Perception Package to improve cross-domain generalization and downstream performance.

While these mitigation strategies demonstrate measurable downstream performance benefits, the reality gap and diversity constraints remain a central limitation of game engine synthetic data, impacting the transferability of CV models to real-world settings. Consequently, recent studies have explored generative modelling for producing realistic synthetic images, an approach examined in the following section.

2.2. Synthetic Dataset Generation Using Generative Models

Introduced in 2014, GANs [60] employ an adversarial training framework in which two neural networks compete in a minimax game to generate realistic synthetic images. Unlike game engines, which rely on manual scene construction, GANs learn underlying data distributions directly from training samples, enabling them to produce sharper and more lifelike outputs than prior generative methods such as variational autoencoders (VAEs) [87]. Since their inception, various GAN architectures have been proposed for image synthesis, with StyleGAN2-ADA emerging as a particularly effective approach due to its incorporation of adaptive discriminator augmentation (ADA), which mitigates overfitting when training on limited datasets. As shown in Table 2, StyleGAN2-ADA has been successfully applied across diverse domains spanning medical diagnostics, environmental monitoring, and infrastructure inspection, demonstrating the versatility and effectiveness of generative modelling for enhancing downstream performance across a range of CV tasks.

For generative models, the FID metric has been widely adopted to quantify the fidelity of generated samples relative to real data, with lower scores typically indicating closer alignment to real-world statistical distributions [88]. For example, [89] demonstrated high quantitative fidelity using StyleGAN2-ADA to synthesize images of skin lesions for melanoma classification, achieving

 doi:10.20944/preprints202603.0954.v1

an FID score of 0.79. Similarly, studies such as [90–92] produced anatomically realistic magnetic resonance imaging (MRI) scans, with FID values ranging from 18.14 to 67.53. Despite this, recent studies have scrutinized the reliability of FID due to inherent limitations such as biased estimations and incorrect distributional assumptions [93,94]. Notably, dental radiographs synthesized in [95] were rated indistinguishable from real scans by domain experts, yet they achieved an FID of 72.76, the highest recorded in Table 2. Conversely, despite obtaining an FID score of 20.90, chest X-rays produced in [96] contained artefacts that hindered downstream classification performance. These findings highlight that FID may not necessarily reflect the practical utility of synthetic images for downstream tasks, and that quantitative metrics alone may be insufficient to assess generation quality.

Dataset size is also closely tied to the effectiveness of generative models, with insufficient data often leading to discriminator overfitting and training instability [62]. For instance, [97] found that over 10 000 training images were required to synthesize realistic petrographic samples. Similarly, [89] reported improved image quality when training with datasets exceeding 30 000 images, demonstrating the correlation between dataset size and synthetic image fidelity. Conversely, studies utilizing fewer than 2 000 images [90,95] generally produced lower-quality outputs with inconsistent results, underscoring the significance of adequate training data. Despite this general trend, synthesis quality can also be influenced by domain complexity, independent of dataset size. For example, [98] generated realistic pavement crack images using only 778 training samples, while [91] achieved high-fidelity abdominal MRI scan synthesis with 1 300 images. In contrast, [99] obtained an FID of 67.47 when training on 770 landslide images, likely due to the visual diversity inherent to environmental images. These findings suggest that while larger datasets generally enhance synthesis quality, the amount of data required to achieve high-fidelity generation varies considerably across domains depending on visual complexity.

To address these data requirements, recent studies have explored transfer learning strategies whereby pre-trained models are fine-tuned using small amounts of real-world data from the target domain [100]. For instance, [92] demonstrated that pre-training StyleGAN2-ADA on unrelated source domains such as FFHQ [101] improves synthesis quality for brain tumor MRI scans. Similarly, [95] reported that transfer learning not only improved FID scores of synthesized dental radiographs but also enhanced the accessibility of generative modelling for researchers with limited access to computational resources such as GPUs. Collectively, these studies indicate that transfer learning offers a promising yet largely unexplored avenue for addressing data scarcity in generative modelling [92,95,100,102].

While transfer learning may partially alleviate the challenges of data scarcity, generative models fundamentally require substantial quantities of training data to learn meaningful distributions. Even StyleGAN2-ADA, designed specifically for data-constrained scenarios, typically requires thousands of images to produce high-fidelity outputs [62], presenting a significant barrier in domains where data acquisition is limited by cost or accessibility, thereby motivating the hybrid framework for synthetic data generation introduced in the following section.

Table 2. Summary of related works on synthetic dataset generation using StyleGAN2-ADA.

Application	Downstream Task	Training Dataset	Training Configuration	FID Score
Petrographic image classification [97]	Classification	10 070 real petrographic images	6 520 kimg, NVIDIA Quadro RTX 5000	12.49
Brain tumor classification [92]	Classification	3 064 real brain scans	NVIDIA Tesla P100	58.11 – 67.53

 doi:10.20944/preprints202603.0954.v1

Abdominal scan synthesis [91]	Not reported	1 300 real abdominal scans	7 800 kimg, NVIDIA GeForce RTX 2080	18.14
Algal bloom detection [102]	Semantic segmentation	3 114 real algal bloom images	NVIDIA Tesla P100	42.56
Dental radiograph classification [95]	Classification	1 456 real dental radiographs	NVIDIA Tesla A100	72.76
Brain scan synthesis [90]	Not reported	1 412 real brain scans	1 800 kimg, NVIDIA Tesla A100	20.21
Chest X-ray classification [96]	Classification	3 616 real chest X-rays	NVIDIA Tesla K80	20.90
Skin cancer classification [89]	Classification	33 126 real skin lesion images	NVIDIA GeForce RTX 3090	0.79
Landslide detection [99]	Semantic segmentation	770 real landslide images	Not reported	67.47
Wildfire detection [103]	Object detection	1 865 real wildfire images	25 000 kimg, NVIDIA GeForce RTX 3090 Ti	24.07
Pavement crack detection [98]	Semantic segmentation	778 real crack images	32 000 kimg, NVIDIA Tesla T4	6.30

3. Materials and Methods

This study proposes a hybrid synthetic dataset generation framework that integrates game engine rendering with generative modelling to address data scarcity in open-pit crack detection. Leveraging UE5 and StyleGAN2-ADA, the framework synthesizes diverse and realistic images of surface cracks that are automatically annotated using Grounding DINO [104] to train the YOLOv11 real-time object detection model. The methodology comprises three primary stages, as outlined in Figure 2.

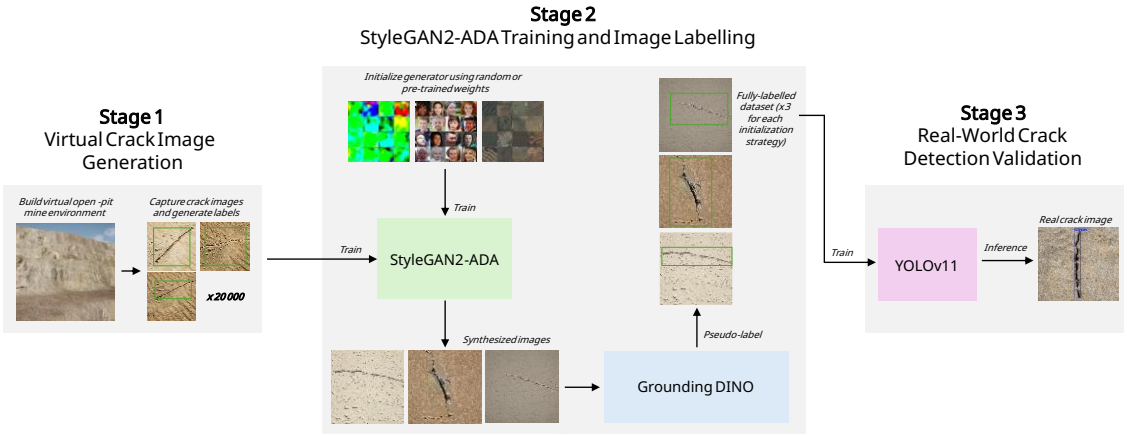


Figure 2. Overview of the proposed hybrid synthetic dataset generation framework. Stage 1 constructs a virtual open-pit mine environment in UE5 and automatically captures crack images with ground-truth bounding boxes. These labelled UE5 images are then used to train StyleGAN2-ADA in Stage 2, where crack images are generated from sampled latent noise or pre-trained weights and subsequently pseudo-labelled using Grounding DINO to

 doi:10.20944/preprints202603.0954.v1

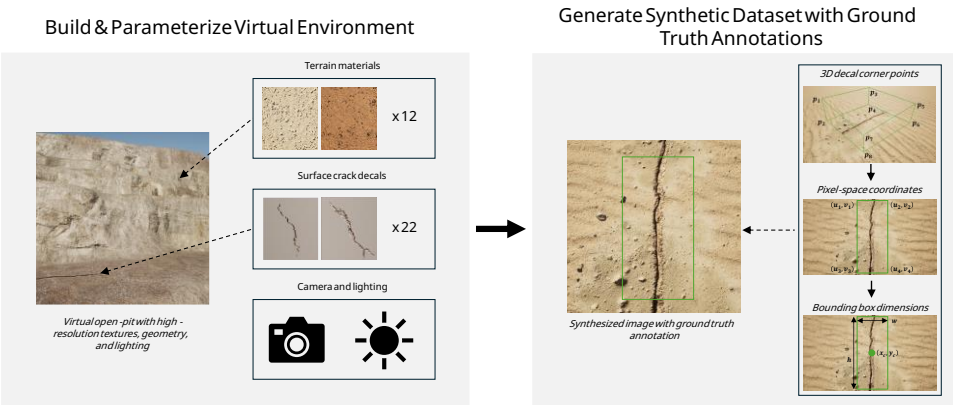
produce bounding boxes for the synthesized samples. In Stage 3, YOLOv11 is trained exclusively on these synthetic datasets and tested on real-world imagery to assess the effectiveness of the proposed pipeline in improving surface crack detection performance for data-scarce open-pit mining.

The first stage (Section 3.1) focuses on the development of a parameterized UE5 environment. A virtual scene representing an open-pit mine wall is first constructed, followed by the development of an automated dataset generation algorithm for domain randomization and data acquisition. By systematically varying crack meshes, ground material textures, and lighting parameters, this algorithm generates a dataset of 20 000 labelled images of open-pit surface cracks. In the second stage (Section 3.2), StyleGAN2-ADA is trained on the UE5 dataset to overcome the diversity ceiling of parametric rendering, generating crack images with structural variations not present in the original UE5 dataset. Three initialization strategies are evaluated to generate 20 000 images per configuration to identify the influence of transfer learning. Each generated image is automatically annotated using Grounding DINO, a vision language model (VLM) capable of detecting objects from a text prompt and generating bounding boxes for downstream CV model training. In the final stage (Section 3.3), a dataset-level ablation study is conducted to assess how each dataset configuration influences downstream YOLOv11 crack detection performance. Model accuracy is assessed on 200 real-world mining images across key performance metrics such as AP, precision, recall, and F1 score, to quantify the generalization capability of object detection models trained solely on game engine data and those trained on synthetic data generated by the proposed framework.

Together, these three stages form a unified framework that synthesizes training data through game engine rendering, expands dataset scale and diversity through generative modelling, and trains generalizable real-time object detection models, thereby enabling autonomous hazard identification to improve operational safety in data-scarce open-pit mining while minimizing manual data collection and annotation effort.

3.1. Synthetic Dataset Generation Using UE5

We adopt UE5 for dataset synthesis due to its state of the art (SOTA) physically based rendering (PBR) pipeline with support for virtualized geometry [105] and real-time global illumination (RTGI) [106]. These features enable the generation of high-fidelity terrain surfaces and illumination effects, thereby reducing the impacts of the reality gap and allowing for more robust downstream generalization [80]. We first construct a configurable virtual open-pit environment (Section 3.1.1) wherein surface cracks are rendered and photographed under systematically randomized conditions. Parameters such as surface appearance and texture (Section 3.1.2), crack morphology (Section 3.1.3), illumination (Section 3.1.4), and camera viewpoint (Section 3.1.5) are independently varied to enhance dataset diversity. High-resolution images (Section 3.1.6) and corresponding bounding box annotations (Section 3.1.7) are automatically generated using a dataset generation pipeline (Section 3.1.8) prior to downstream generative modelling using StyleGAN2-ADA (Section 3.2). A high-level overview of the complete workflow is shown in Figure 3.



 doi:10.20944/preprints202603.0954.v1

Figure 3. High-level overview of the UE5 synthetic dataset generation pipeline. A parameterized open-pit mine environment is constructed using virtualized geometry exhibiting realistic benches and haul roads, slope faces, and weathered rock surfaces. Domain randomization is applied across variables such as illumination, camera parameters, surface appearance, and crack decal morphology. For each randomized scene instance, high resolution images are captured and annotated by projecting the 3D decal corner points into image space to compute ground-truth bounding box dimensions. Image and label pairs are then exported for downstream generative modelling using StyleGAN2-ADA.

3.1.1. Virtual Environment Construction

A virtualized section of an open-pit wall is constructed using the Landscape tool [107] in UE5, as illustrated in Figure 3. Designed to support photorealistic visual rendering rather than explicit geotechnical modelling, the generated heightmap incorporates representative slope surfaces and surrounding context largely consistent with operational open-pit settings, without attempting to reproduce site-specific stratigraphy or failure mechanics. The primary objective of this 3D scene is to provide a visually realistic background surface against which surface cracks are rendered, and for this reason, emphasis is placed on surface texture, color variation, and roughness, an approach consistent with domain randomization methodologies commonly adopted in synthetic data generation to enhance model generalization across unseen settings [108,109].

3.1.2. Terrain Material Parameterization

To effectively reflect the visual diversity typical of open-pit mine sites, the generated landscape is parameterized using 12 distinct terrain surface materials selected to span the range of surface conditions typically documented across Australian mining operations [110,111]. These materials represent commonly observed surface types including soils, sandy deposits, compacted gravel, weathered rock formations, ironstone, and mixed debris zones, with color variations spanning red, ochre, grey, and brown tones, as demonstrated in Figure 3. Each high-resolution material, sourced from Quixel Megascans [112], incorporates PBR properties including albedo, normal, and roughness, as well as Nanite [105] displacement maps to support realistic light interaction. By randomly varying surface appearance while maintaining fixed terrain geometry during dataset generation, controlled background variability is introduced to improve model robustness under domain shift, allowing for enhanced generalizability across different operational environments.

3.1.3. Surface Crack Decal Parameterization

Surface crack decals are generated from real-world crack imagery to preserve authentic morphological characteristics. Crack images acquired from site surveys are first manually processed to extract binary crack masks. These masks are then converted into grayscale opacity maps with a resolution of 2048 × 2048 pixels prior to additional processing to generate auxiliary texture maps such as normal, height, and roughness for enhanced decal realism. These textures are then imported into UE5 and assembled into deferred decal materials, forming the unique cracks demonstrated in Figure 3. A total of 22 crack decal variants are generated, representing common patterns such as single cracks, bifurcated cracks, and crossed cracks [113]. During instantiation in UE5, decals are scaled anisotropically to preserve field-measured aspect ratios, ensuring geometric consistency with real-world crack morphology.

3.1.4. Lighting Positioning and Intensity Parameterization

Lighting parameters are independently randomized per Figure 3 to replicate the diverse conditions encountered during site inspections. The directional light component, representing solar illumination, is configured with continuously varying position and intensity. Azimuth angle θ spans full 360° rotation along the horizontal plane, while elevation angle ψ varies between 30° and 90° to represent solar position changes from early morning to overhead midday conditions. Light intensity

 doi:10.20944/preprints202603.0954.v1

I ranges from 5 to 10 lux with a fixed color temperature of 5000 K to represent daylight white balance. This systematic randomization of solar position and intensity produces natural changes in shadow direction and surface contrast, helping prevent overfitting and improving model robustness to real-world lighting variability.

3.1.5. Camera Viewpoint Parameterization

Virtual camera positioning employs randomized spherical coordinate parameterization relative to the center of the instantiated crack, thereby ensuring coverage of geotechnically relevant inspection viewpoints. The standoff distance d is sampled between 1 and 10 m to represent the typical range for both ground-based and unmanned aerial vehicle (UAV) inspection [114]. The azimuth angle ϕ provides 360° rotation for complete directional coverage around the crack, eliminating bias towards specific viewing directions, while the elevation angle α varies between 45° and 90° to span oblique to nadir perspectives. Slight camera jitter ϱ between -10° and 10° across pitch and roll axes simulates natural handheld tilt and UAV attitude changes, whereas field of view (FOV) variations from 70° to 110° represent the typical range of smartphone and digital single-lens reflex (DSLR) cameras. Focus distance is automatically set to match the standoff distance, ensuring consistent sharpness across viewpoints. Additional intrinsic camera parameters for the Cine Camera Actor component [115] are configured as summarized in Table 3.

Table 3. Cine Camera Actor settings used for synthetic dataset generation in UE5.

Setting	Value
Sensor Format	36 mm × 20.25 mm
Aspect Ratio	16:9
Resolution	1920 × 1080 pixels
Aperture	f/5.6
ISO	100
Shutter Speed	1/500 s

3.1.6. Synthetic Image Rendering

For each randomized scene instance, a 1920 × 1080 resolution image is rendered using the High Resolution Screenshot Tool (HRSST) [116]. Global illumination is enabled via Lumen [106] in hardware ray tracing (RT) mode with high-quality settings to achieve photorealistic lighting behavior. Temporal anti-aliasing (TAA) is used to suppress spatial aliasing, while post-processing effects such as auto-exposure, chromatic aberration, vignetting, film grain, and lens distortion are disabled to maintain consistent image quality across captures.

3.1.7. Bounding Box Computation

As illustrated in Figure 3, each crack decal actor is approximated by a 3D box with center p_c and half-extents e_x, e_y, and e_z, with corresponding corner points defined as:

$$p_i = p_c + (\pm e_x, \pm e_y, \pm e_z), \qquad i = 1, \ldots, 8. \tag{1}$$

To transform these 3D world coordinates into 2D image space for bounding box computation, each corner point is projected to pixel space using the camera projection operator $\Pi(\cdot)$:

$$(u_i, v_i) = \Pi(p_i), \tag{2}$$

where u_i and v_i represent the horizontal and vertical pixel coordinates of the i-th projected corner clamped to the viewport bounds $[0, W] \times [0, H]$, with $W = 1920$ and $H = 1080$.

The operator $\Pi(\cdot)$ performs perspective division and maps world coordinates to pixel coordinates using a 4 × 4 reversed-Z perspective projection matrix P:

 doi:10.20944/preprints202603.0954.v1

$$P = \begin{bmatrix} \frac{1}{\tan(\alpha/2)} & 0 & 0 & 0 \\ 0 & \frac{W}{H \cdot \tan(\alpha/2)} & 0 & 0 \\ 0 & 0 & \frac{n}{n-f} & 1 \\ 0 & 0 & -\frac{f \cdot n}{n-f} & 0 \end{bmatrix}, \quad (3)$$

where α represents the vertical FOV half-angle (the angle from the center of the lens to the edge of the viewable area), and n and f correspond to the near and far clipping planes, respectively [117].

From the set of eight projected corner points $\{(u_i, v_i)\}$, the enclosed 2D bounding box coordinates are computed as:

$$u_{min} = \min_i u_i, \quad u_{max} = \max_i u_i, \quad v_{min} = \min_i v_i, \quad v_{max} = \max_i v_i. \quad (4)$$

Pixel-space center coordinates and bounding box dimensions w and h are then calculated as:

$$x_c{}^{px} = \frac{u_{min} + u_{max}}{2}, \quad y_c{}^{px} = \frac{v_{min} + v_{max}}{2}, \quad w^{px} = u_{max} - u_{min}, \quad h^{px} = v_{max} - v_{min}. \quad (5)$$

These quantities are then normalized with respect to the viewport dimensions W and H and exported as a label file in the YOLO annotation format [118]:

$$x_c = \frac{x_c{}^{px}}{W}, \quad y_c = \frac{y_c{}^{px}}{H}, \quad w = \frac{w^{px}}{W}, \quad h = \frac{h^{px}}{H}. \quad (6)$$

3.1.8. Automated Dataset Generation Pipeline

Automated dataset generation is implemented through the UE5 Blueprint script presented in Algorithm 1, which iteratively randomizes crack decals, terrain materials, lighting conditions, and camera geometry for each rendered image using the distributions defined in Sections 3.1.2 – 3.1.5. For each rendered image, the corresponding annotation is computed as described in Section 3.1.7 to produce a total of 20 000 labelled synthetic surface crack images. Dataset generation is performed using an NVIDIA GeForce RTX 5090 GPU, 64 GB RAM, and Intel Core Ultra 9 285H CPU in UE5 version 5.6.1.

Algorithm 1. Automated Synthetic Dataset Generation Pipeline for UE5

Input: Dataset size N, Crack decals C, Terrain materials T

Output: Images $I = \{I_1, \ldots, I_N\}$, Labels $L = \{L_1, \ldots, L_N\}$

for $i = 1$ to N **do**
  // Sample crack decal and terrain material
  crack ← SampleCrack(C), terrain ← SampleMaterial(T)
  // Sample lighting and camera parameters
  (θ, ψ, I) ← SampleLighting(), (d, ϕ, α, ϱ, FOV) ← SampleCamera()
  // Configure scene with sampled parameters
  SpawnDecal(C), ApplyTerrainMaterial(T), SetDirectionalLight(θ, ψ, I)
  // Compute and assign camera position
  position ← SphericalToCartesian(d, ϕ, α)
  SetPosition(position, roll = ϱ, perspective = FOV)
  // Render image and compute annotation
  I_i = CaptureImage(), L_i = ComputeBoundingBox()
  // Prepare for next iteration
  DestroyDecal(C)
end for
return I, L

 doi:10.20944/preprints202603.0954.v1

3.2. Synthetic Dataset Diversity Enhancement Using StyleGAN2-ADA

Using the synthetic images generated in UE5, we train StyleGAN2-ADA (Section 3.2.1) to increase the diversity of our dataset by generating realistic images of surface cracks with structural variations not captured in the UE5 dataset. To investigate the effect of different initialization strategies, three transfer learning configurations (Section 3.2.2) are evaluated with respect to generation fidelity, quantified using FID (Section 3.2.3), and generation diversity, measured using LPIPS (Section 3.2.4). The resulting 20 000 images generated for each configuration are then automatically annotated with bounding boxes using Grounding DINO and filtered using confidence thresholding and visual inspection (Section 3.2.5).

3.2.1. StyleGAN2-ADA Architecture Overview

We adopt the StyleGAN2-ADA architecture, which is capable of generating high-fidelity synthetic images even under data-scarce conditions. The architecture consists of a generator, which includes mapping and synthesis networks that produce synthetic images, and a discriminator, which evaluates the realism of generated samples, as illustrated in Figure 4. The mapping network contains a multilayer perceptron (MLP) with eight fully connected (FC) layers which transform an input vector in latent space $\mathbf{z} \in Z$ into an intermediate latent code $\mathbf{w} \in W$. The synthesis network then produces images through a hierarchy of style-modulated convolution blocks with learned weight demodulation spanning multiple resolution scales. Per-layer affine transforms **A** from **w** produce the style parameters that control the convolutions performed in each of these style blocks, thereby controlling visual attributes and image characteristics, while injected stochastic noise **B** provides finer, unstructured detail. The discriminator employs a multi-stage residual architecture, downsampling full-resolution inputs and passing the final feature map through a FC layer to produce a single scalar output $D(x)$ representing the probability that an input image is real or generated. To maintain training stability under limited data conditions, ADA dynamically applies random geometric and color-space perturbations to discriminator inputs to prevent overfitting while preserving generator output diversity.

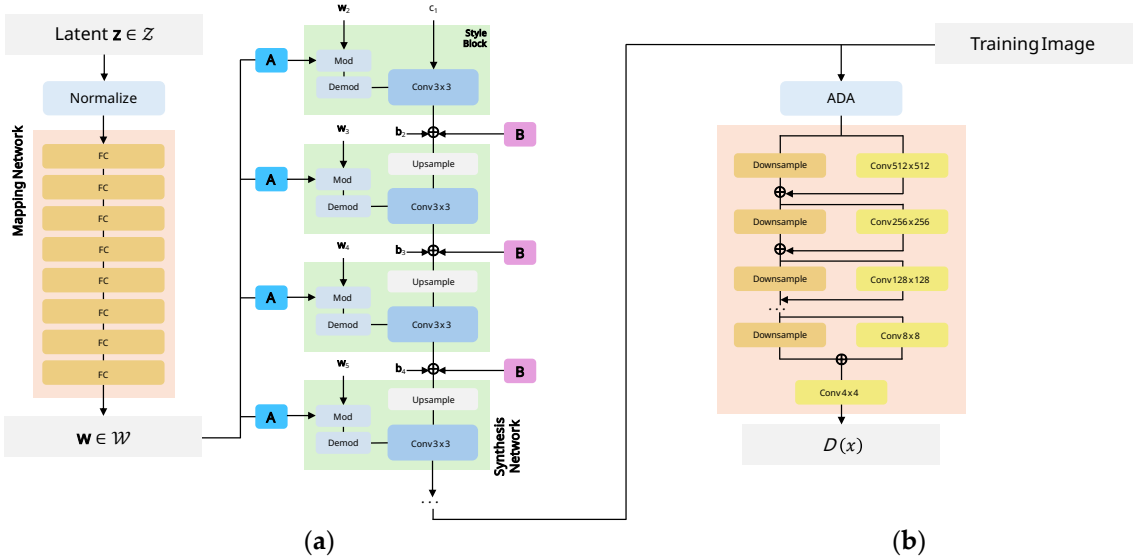


Figure 4. Architecture of StyleGAN2-ADA, comprising (**a**) the generator and (**b**) the discriminator. The mapping network transforms a latent vector $\mathbf{z} \in Z$ into an intermediate latent representation $\mathbf{w} \in W$, which modulates the synthesis network through per-layer affine transforms **A**. The synthesis network starts from a constant input c_1 and progressively refines features using modulated style blocks and stochastic noise **B** to introduce unstructured detail. The discriminator uses ADA to apply random geometric and color-space perturbations to real and generated images, mitigating overfitting under limited-data conditions. Progressive downsampling with

 doi:10.20944/preprints202603.0954.v1

residual connections produces a scalar output $D(x)$ representing the probability that an image is real or generated.

3.2.2. StyleGAN2-ADA Training Configuration

As discussed throughout Section 2.2, generative models typically require tens of thousands of training images to produce high-fidelity outputs [62,103], with techniques such as transfer learning potentially enhancing their generalizability in data-constrained settings [92,95,100]. Pre-training leverages visual priors learned from a source domain to accelerate convergence and improve generation quality in a target domain. To examine the impact of transfer learning on the synthesis of surface crack images using StyleGAN2-ADA, three training configurations are developed to represent distinct points along a domain similarity spectrum.

The first configuration trains StyleGAN2-ADA solely on game engine data, serving as a baseline for quantifying the generation quality achievable without external knowledge transfer. Both the generator and discriminator are initialized with random weights and trained for 2 000 kimg on the UE5 dataset.

The second configuration leverages Flickr-Faces-HQ (FFHQ) [101], pre-trained on 70 000 high-resolution images of human faces, to examine the impact of transfer learning from a semantically unrelated source domain. This configuration is motivated by prior research demonstrating that low-level visual features, such as edge and texture primitives, exhibit strong transferability across semantically different domains [92]. The pre-trained model is fine-tuned using the UE5 dataset for 2 000 kimg.

The third configuration utilizes weights pre-trained on the Describable Textures Dataset (DTD) [119], comprising 5 640 images across 47 texture categories at 1024 × 1024 resolution [120]. Unlike the FFHQ dataset, DTD contains, amongst other categories, explicit representations of cracked, fractured, and rough surface patterns, providing strong low-level visual correspondence between source and target domains. Fine-tuned using the UE5 dataset for 2 000 kimg, this configuration enables assessment of the impact of explicit texture-focused pre-training on the generation of surface crack images.

All training configurations use the official NVIDIA StyleGAN3 repository [121] with StyleGAN2-ADA architecture configuration due to improved compatibility with current versions of PyTorch. Training is conducted on an NVIDIA Tesla A100 GPU with CUDA 12.8, PyTorch 2.2.0, and Python 3.10. Training images are downsampled from 1920 × 1080 to 512 × 512 resolution to balance spatial detail with computational efficiency and training stability. A batch size of 16 is selected to ensure gradient stability within GPU memory constraints. As crack morphology remains invariant under horizontal reflection, mirror augmentation is enabled to increase training data diversity and improve generalization. Additional training hyperparameters follow the StyleGAN2-ADA default values [122] summarized in Table 4.

Table 4. Default hyperparameter configuration used for training StyleGAN2-ADA.

Hyperparameter	Value
Learning Rate	0.002
Optimizer	Adam ($\beta_1 = 0$, $\beta_2 = 0.99$, $\varepsilon = 1e^{-8}$)
R1 Regularization Weight	10.0
Effective R1 Weight	160
Path Length Regularization Interval	4 iterations
R1 Regularization Interval	16 iterations
ADA Target	0.6 (60%)
Loss Function	Non-saturating logistic loss

 doi:10.20944/preprints202603.0954.v1

3.2.3. Generation Fidelity Evaluation

We employ the FID [123] to assess generation fidelity, a metric which measures the distributional similarity between real and generated images in the feature space of a pre-trained Inception-v3 network [124]. This metric quantifies the distance between the feature distributions of real and generated images as follows:

$$d^2\big((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)\big) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{T_r}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2(\boldsymbol{C}\boldsymbol{C}_w)^{1/2}\big), \quad (7)$$

where ($\boldsymbol{m}$, $\boldsymbol{C}$) and ($\boldsymbol{m}_w$, $\boldsymbol{C}_w$) denote the mean vectors and covariance matrices of the real and generated image features extracted from the Inception-v3 activation space respectively, and T_r represents the matrix trace. Lower FID scores generally indicate a greater similarity between the Gaussian approximations of real and generated feature distributions p and p_w, thereby indicating higher generation fidelity.

To monitor generation quality throughout training, FID scores are computed on selected checkpoints (every 200 kimg) spanning the training duration by synthesizing 50 000 random samples and comparing their feature distributions against 50 000 images sampled with replacement from the training set. This temporal tracking facilitates analysis of convergence behavior, typically marked by FID score stabilization. Upon convergence and completion of training, the best checkpoint for each of the three configurations is selected based on the lowest FID score achieved using full latent space sampling to ensure unbiased evaluation. Finally, 20 000 surface crack images are generated using the best checkpoint for each training configuration through random sampling of the latent space.

3.2.4. Generation Diversity Evaluation

We evaluate generation diversity by utilizing LPIPS [125], a metric which measures perceptual distance aligning closely with human visual judgement by quantifying how far apart two images are in the feature space of a pre-trained network. As shown in Equation 8, LPIPS computes this perceptual distance by measuring the squared l_2 difference between the feature activations y^l and $y^l{}_0$ of two images across multiple layers of a network F, in our case, Visual Geometry Group-16 (VGG-16) [126]. These feature differences are unit-normalized and scaled by vector $\boldsymbol{w}^l$ prior to being averaged spatially and summed channel-wise, producing a scalar output d that correlates strongly with human judgements of visual similarity. Lower scores, typically close to 0, indicate greater perceptual similarity, whereas higher scores suggest that image pairs look more different and diverse to humans.

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| \boldsymbol{w}^l \odot \left(\hat{y}^l{}_{hw} - \hat{y}^l{}_{0hw} \right) \right\|_2^2. \quad (8)$$

To assess whether StyleGAN2-ADA has generated images of surface crack structures with greater diversity than those present in the original UE5 training dataset, we conduct an intra-set diversity comparison by analyzing both mean LPIPS and clustering in the VGG-16 feature space. To do so, we randomly sample 5 000 pairs of images from each of our synthesized datasets and calculate the distance between each pair. By comparing their statistical distributions, namely the mean and median LPIPS, we are able to determine whether StyleGAN2-ADA has produced datasets with enhanced diversity. We also extract the feature vectors from our datasets using the VGG-16 backbone and use t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the image embeddings. The number and separation of clusters in t-SNE space provides qualitative insight into distributional coverage, where a greater number of separated clusters suggests broader perceptual variability and therefore, increased dataset diversity.

3.2.5. Automated Annotating Using Grounding DINO

Manual annotation in CV has long been a time-consuming and resource-intensive task. To address this challenge, we employ pseudo-labelling using Grounding DINO [104], a unified VLM designed for open-set object detection due to its ability to identify objects it has never explicitly seen

 doi:10.20944/preprints202603.0954.v1

during training using language-guided text prompts. This zero-shot inference approach allows the model to generate labels for previously unseen data, such as the images synthesized by StyleGAN2-ADA. Grounding DINO adopts a transformer-based encoder-decoder architecture to process visual features extracted from images and fuse them with prompt information via cross-attention. The image backbone, a pre-trained ViT, such as Swin Transformer [127], is used to extract multi-scale visual features from input images, while a BERT-based [128] text prompt encoder converts prompt tokens into semantic embeddings that guide the detection process. The model integrates these representations within a feature enhancer module, producing language-conditioned object queries that are passed to the decoder, whose outputs are directly projected into bounding-box coordinates and text-region alignment scores through prediction layers. We utilize Grounding DINO in a zero-shot manner with the text prompt "crack" to generate a set of bounding boxes with confidence scores for each of our StyleGAN2-ADA images. Low-confidence predictions with scores below 0.7 are automatically removed via confidence thresholding and the remaining annotations are manually inspected prior to being converted to YOLO format, yielding a large-scale pseudo-labelled dataset suitable for downstream object detection model training. An overview of this automated dataset annotation pipeline is shown in Figure 5.

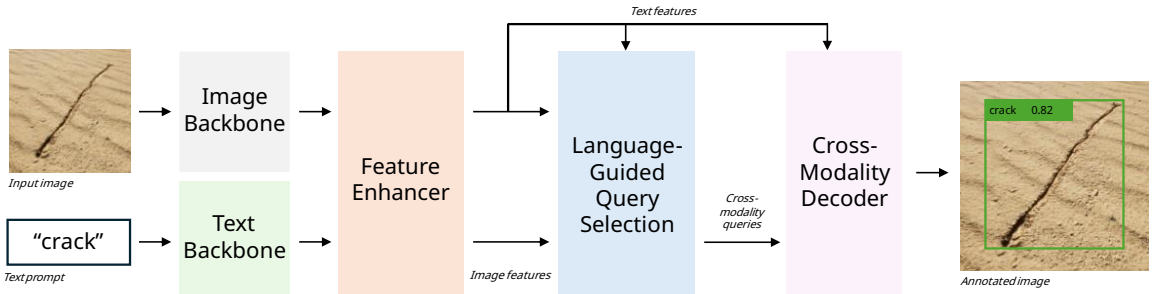


Figure 5. High-level overview of the Grounding DINO pipeline used for pseudo-labelling the images generated by StyleGAN2-ADA. The input image and text prompt are encoded by their respective backbones and fused in the feature enhancer, which injects semantic text information into the visual features via cross-attention. The resulting language-conditioned object queries, together with multi-scale image features, are processed by the cross-modality decoder. The decoder outputs are then linearly projected into bounding-box coordinates and text-region alignment scores, yielding detections corresponding to the text prompt used, in this case, "crack".

3.3. Crack Detection Using YOLOv11

3.3.1. YOLOv11 Architecture Overview

We utilize YOLOv11 [129] to evaluate the downstream effectiveness of the datasets synthesized by the proposed framework. This one-stage CNN-based real-time object detection model improves upon prior iterations through architectural refinements that enhance multi-scale feature extraction, reduce computational overhead, and increase overall robustness, making it well-suited to real-world operational contexts. As illustrated in Figure 6, YOLOv11 follows a modular design comprising three principal components: a backbone, a neck, and a detection head. The backbone consists of stacked CBS and C3K2 blocks that progressively reduce the spatial resolution of input images while increasing channel depth, enabling efficient extraction of features such as crack width and curvature. It also incorporates the spatial pyramid pooling-fast (SPPF) block to expand the receptive field and combine multi-scale contextual information, and C2PSA attention modules to more effectively capture both local and global features, thereby improving detection accuracy across objects of varying scales. The neck facilitates multi-scale feature fusion through a series of upsampling, downsampling, and concatenation operations that refine information from different stages of the backbone, enhancing robustness to scale variation. Finally, the detection head employs decoupled classification

 doi:10.20944/preprints202603.0954.v1

and regression branches, enabling more precise object localization and bounding box prediction by processing the fused features transmitted from the neck.

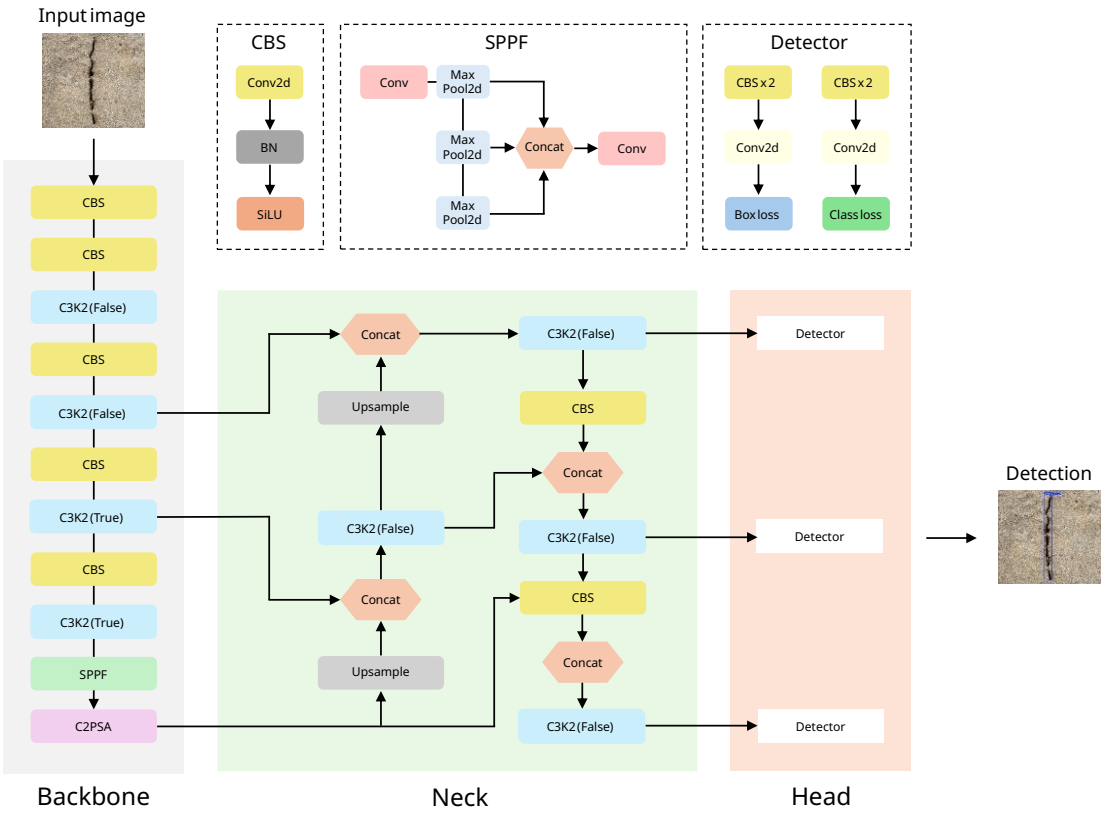


Figure 6. Architectural overview of YOLOv11, comprising a backbone, neck, and detection head for real-time crack detection. The backbone uses stacked CBS and C3K2 blocks to progressively downsample the input while increasing channel depth to extract features relevant to crack morphology. The SPPF block enlarges the receptive field by combining multi-scale context, and C2PSA modules enhance feature representation through spatial and channel attention. The neck performs multi-scale feature fusion via upsampling, downsampling, and concatenation operations that integrate information from different backbone stages. The decoupled detection head then applies concurrent classification and regression branches to generate bounding box coordinates and class scores for detected cracks.

3.3.2. YOLOv11 Training Configuration

To determine whether the proposed framework provides measurable benefits for surface crack detection, we conduct a dataset-level ablation study using YOLOv11. Four dataset variants are evaluated: (i) UE5 only (UE5O), representing the baseline; (ii) UE5O augmented with StyleGAN2-ADA (SG2), used to assess whether the proposed framework improves detection performance beyond UE5O; (iii) SG2 with FFHQ pre-training (SG2-FFHQ); and (iv) SG2 with DTD pre-training (SG2-DTD). These configurations, summarized in Table 5, are designed to evaluate the impact that generative refinement of synthetic data has on downstream object detection performance. Training YOLOv11 on each dataset separately also provides a controllable basis for isolating the effects of StyleGAN2-ADA initialization strategies on real-world generalization performance.

Table 5. Dataset configurations used for training YOLOv11 for open-pit surface crack detection.

Dataset	Total Images	Training Images	Validation Images
UE5O	20 000	17 000	3 000
UE5O + SG2	40 000	34 000	6 000

 doi:10.20944/preprints202603.0954.v1

UE5O + SG2-FFHQ	40 000	34 000	6 000
UE5O + SG2-DTD	40 000	34 000	6 000

Training is conducted using the official YOLOv11 implementation on an NVIDIA Tesla A100 GPU with CUDA 12.8, PyTorch 2.2.0, and Python 3.10. The medium model variant (YOLOv11m) is selected to balance detection performance and computational efficiency. COCO pre-trained weights are used for the UE5O configuration, while SG2-adapted models are fine-tuned from the best-performing UE5O weights. All remaining training hyperparameters are reported in Table 6.

Table 6. Hyperparameter configuration used for training YOLOv11.

Hyperparameter	Value
Model	YOLOv11m
Initialization Weights	COCO
Input Resolution	512 × 512 pixels
Batch Size	64
Epochs	300
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.99$)
Initial Learning Rate	0.001
Learning Rate Schedule	Cosine decay
Warmup Epochs	3
Weight Decay	0.0005
Data Augmentation	On (scaling, translation, flip, mosaic)

3.3.3. Object Detection Performance Evaluation

Model performance is evaluated using a held-out test set of 200 real-world mining images (see Figure 7) collected from various open-pit mining operations across Australia using UAVs and handheld cameras. The test set exhibits considerable variability in lighting conditions, camera viewpoint, and environmental context, enabling a rigorous assessment of model generalization beyond synthetic data distributions.

Figure 7. Examples of real-world open-pit mine surface crack images used for YOLOv11 performance evaluation.

Model performance is quantified using precision, recall, F1 score, and AP evaluated at different IoU thresholds. In this study, a predicted bounding box is considered a true positive (TP) if its overlap with a ground-truth bounding box exceeds an IoU threshold of 0.5. Predictions below this threshold are classified as false positives (FP), while ground-truth objects without matching predictions constitute false negatives (FN). From these quantities we compute precision (P), representing the proportion of predicted detections that correctly identify actual surface cracks:

$$P = \frac{TP}{TP + FP} \tag{9}$$

We also calculate recall (R), indicating the proportion of actual surface cracks successfully detected by the model:

 doi:10.20944/preprints202603.0954.v1

$$R = \frac{TP}{TP + FN} \tag{10}$$

We leverage the F1 score as a balanced indicator of detection performance, providing a single measure that reflects model performance with respect to both consistent and accurate crack identification:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{11}$$

Finally, the AP metric, corresponding to the area under the precision-recall curve, is used to provide a consolidated measure of detection quality by integrating precision and recall across all confidence thresholds:

$$AP = \int_0^1 P(R)\, dR \tag{12}$$

We report AP@0.5 (IoU $\geq$ 0.5) and AP@[0.5:0.95] to quantify both detection accuracy and spatial extent localization, two attributes essential for effective surface crack monitoring.

4. Results and Discussion

This section evaluates the effectiveness of the proposed hybrid synthetic dataset generation framework in addressing data scarcity for open-pit surface crack detection. We first examine the convergence characteristics of the StyleGAN2-ADA training configurations, followed by quantitative and qualitative assessments of generation fidelity and diversity (Section 4.1). We then analyze real-world crack detection performance through a dataset-level ablation study using YOLOv11, discussing the impact of different generative model initialization strategies on downstream model robustness, generalizability, and transferability to real-world conditions (Section 4.2). Finally, we reflect on the broader implications of synthetic data for operational deployment of CV models in data-scarce domains, evaluating the practical utility of the proposed framework (Section 4.3). Together, these analyses provide a comprehensive assessment of whether the proposed framework meaningfully improves real-world object detection performance under limited-data conditions.

4.1. StyleGAN2-ADA Training and Image Generation Assessment

4.1.1. Training Dynamics

The training behavior of the StyleGAN2-ADA configurations used for image synthesis in this study are evidenced in Figure 8. As highlighted in Figure 8(a), the baseline configuration (SG2) exhibits higher generator loss at initialization compared to both pre-trained models (SG2 + FFHQ and SG2 + DTD). This behavior is expected when training from scratch, as the generator initially produces unstructured noise, allowing the discriminator to easily classify its outputs as fake samples with a high degree of confidence. SG2 generator loss rapidly decreases within the first 100 kimg as the generator starts to form a coherent latent representation, then decays slowly until reaching convergence of 1.29 around 1 250 kimg. Both pre-trained models, conversely, achieved more rapid convergence with lower initial generator loss due to their already well-structured latent representations. SG2 + FFHQ ultimately achieves the lowest generator loss at 1.13 and converges the most smoothly, indicating that pre-trained weights from the FFHQ-1024 dataset may provide effective feature representations for surface crack image generation, despite originating from a semantically distant and unrelated domain. Domain-aligned pre-training by way of the SG2 + DTD configuration exhibits better initialization and faster convergence than both SG2 and SG2 + FFHQ; however, generator loss fluctuations throughout training suggest comparatively reduced stability relative to the other training configurations.

 doi:10.20944/preprints202603.0954.v1

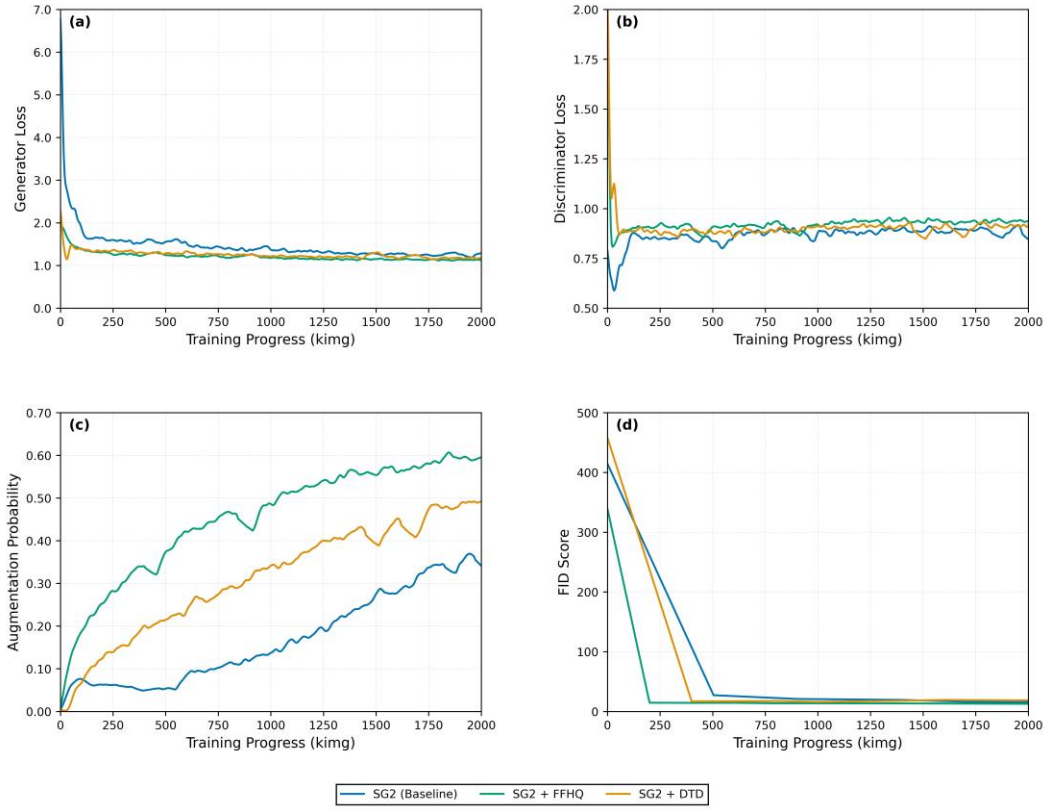


Figure 8. StyleGAN2-ADA training behavior for three initialization strategies. Subplots show (**a**) generator loss, (**b**) discriminator loss, (**c**) augmentation probability, and (**d**) FID score progression over a 2 000 kimg training window for SG2 (Baseline), SG2 + FFHQ, and SG2 + DTD configurations.

While generator loss highlights the impact of initialization and pre-training on generative capacity, discriminator loss, quantified in Figure 8(b), provides complementary insight into the stability and balance of adversarial training dynamics. Discriminator loss across all three configurations converges rapidly within the first 200 kimg and remains tightly clustered for the remainder of training, indicating that the adversarial game stabilizes early and reaches a Nash equilibrium [130]. All three configurations exhibit an initial loss spike caused by the discriminator rapidly adapting to the highly unrealistic outputs synthesized by the generator; however, the discriminator then settles into a narrow range between 0.85 and 0.90 following this transient phase, with only minor fluctuations throughout the full 2 000 kimg training window. SG2 displays slightly lower discriminator loss than the pre-trained configurations during the early stages (< 50 kimg), reflecting the confidence with which the discriminator classifies the random noise synthesized by the uninitialized generator as fake samples. As generator fidelity improves, the discriminator loss rises to match that of the pre-trained configurations, signaling convergence toward a stable equilibrium. Both pre-trained configurations reach this equilibrium more quickly, however, exhibiting nearly indistinguishable trajectories beyond the 100 kimg mark. Overall, the minimal separation between curves indicates that, unlike generator loss, discriminator loss is only weakly influenced by pre-training and instead reflects the balance of the adversarial process once both networks have stabilized, ultimately confirming that all three configurations maintained stable training dynamics without evidence of discriminator collapse.

Figure 8(c) shows the evolution of the augmentation probability applied by the ADA module to prevent discriminator overfitting for each of the three training configurations. This value typically increases when the discriminator exhibits overconfident predictions and decreases when its predictive reliability weakens, and as such, the trajectory of augmentation probability reflects the degree of overfitting pressure placed on the discriminator throughout training. The baseline configuration exhibits the lowest augmentation probability throughout the training run, rising

 doi:10.20944/preprints202603.0954.v1

gradually from near-zero to just 32 %. This behavior indicates that the discriminator experiences comparatively weaker overfitting pressure when the generator is trained from scratch, likely because the generator initially produces low-quality outputs hindered by the limited diversity of the UE5 training dataset. In contrast, both SG2 + DTD and SG2 + FFHQ show markedly higher values of augmentation probability consistent with the discriminator facing higher quality generator outputs. SG2 + DTD rises steadily and plateaus around 50 %, however, SG2 + FFHQ exhibits the highest augmentation probability of 60 %, suggesting that the discriminator is consistently close to overfitting, likely due to the coherent and high-quality outputs of the FFHQ-initialized generator, forcing ADA to inject stronger regularizations to maintain adversarial balance.

Changes in generator fidelity throughout training are demonstrated in Figure 8(d), which illustrates the FID score progression of the three StyleGAN2-ADA initialization strategies. Across all configurations, FID decreases sharply during the first 400 kimg as the network rapidly learns structural and textural characteristics from the training data. Both pre-trained configurations exhibit a steeper initial decline than the baseline configuration, reflecting the advantage conferred by transfer learning through the inheritance of low-level feature priors that improve early-stage generation fidelity. This acceleration in early synthesis quality is further supported by Figure 9, which shows that crack-like structures emerge within the first 30 kimg of training for both pre-trained configurations. In contrast, the baseline configuration requires nearly three times longer to form comparably coherent latent-space structure. Following the initial descent, all curves plateau and show only marginal improvements for the remainder of training, indicating that each configuration collectively reaches a point of diminishing returns relatively early at around 400 kimg. This behavior also confirms that all three configurations achieved stable training dynamics with no evidence of mode collapse, which is often signaled by late-stage FID fluctuations. SG2 + FFHQ ultimately achieves the lowest FID at convergence of 12.75, outperforming both SG2 + DTD (15.99) and SG2 (15.88), further demonstrating that FFHQ pre-training provides the most effective and stable transfer of feature representations for realistic surface crack image synthesis in this study.

These results collectively demonstrate that all three StyleGAN2-ADA configurations converged stably, with the pre-trained configurations, particularly SG2 + FFHQ, achieving faster learning and superior generation fidelity overall. While these training dynamics provide insight into model behavior during optimization, they do not quantify the realism or variability of the images produced by the final configurations. Accordingly, the following section evaluates the fidelity and diversity of the generated samples in greater detail.

4.1.2. Qualitative Evaluation of Generation Fidelity and Diversity

Table 7 provides a quantitative comparison of the final synthesis quality attained by each StyleGAN2-ADA training configuration, evaluated using the FID and LPIPS metrics described in Sections 3.2.3 and 3.2.4. The high quantitative fidelity achieved by SG2 + FFHQ suggests that this configuration effectively generates high-quality surface crack images with greater realism than both SG2 and SG2 + DTD. Notably, all three configurations achieve comparatively excellent FID scores, surpassing many of the results seen in analogous studies such as those documented in Table 2, reinforcing the efficacy of the StyleGAN2-ADA training methodology developed for image synthesis in this study. Qualitative analysis (Section 4.1.3) of the images generated by the StyleGAN2-ADA training configurations is necessary to identify further detail variations encapsulated by the minute changes in FID documented in Table 7.

Table 7. Fidelity (FID), perceptual diversity (LPIPS), and relative LPIPS improvement over the UE5O baseline (LPIPS vs UE5O) for the three StyleGAN2-ADA initialization strategies.

Configuration	FID Score	Mean LPIPS	Median LPIPS	LPIPS Range	LPIPS vs UE5O
SG2 (Baseline)	15.99	0.452 ± 0.094	0.456	[0.011, 0.725]	+ 2.80 %
SG2 + FFHQ	**12.75**	**0.472 ± 0.090**	0.477	[0.140, 0.709]	**+ 7.49 %**
SG2 + DTD	15.88	0.457 ± 0.092	0.462	**[0.115, 0.735]**	+ 3.96 %

 doi:10.20944/preprints202603.0954.v1

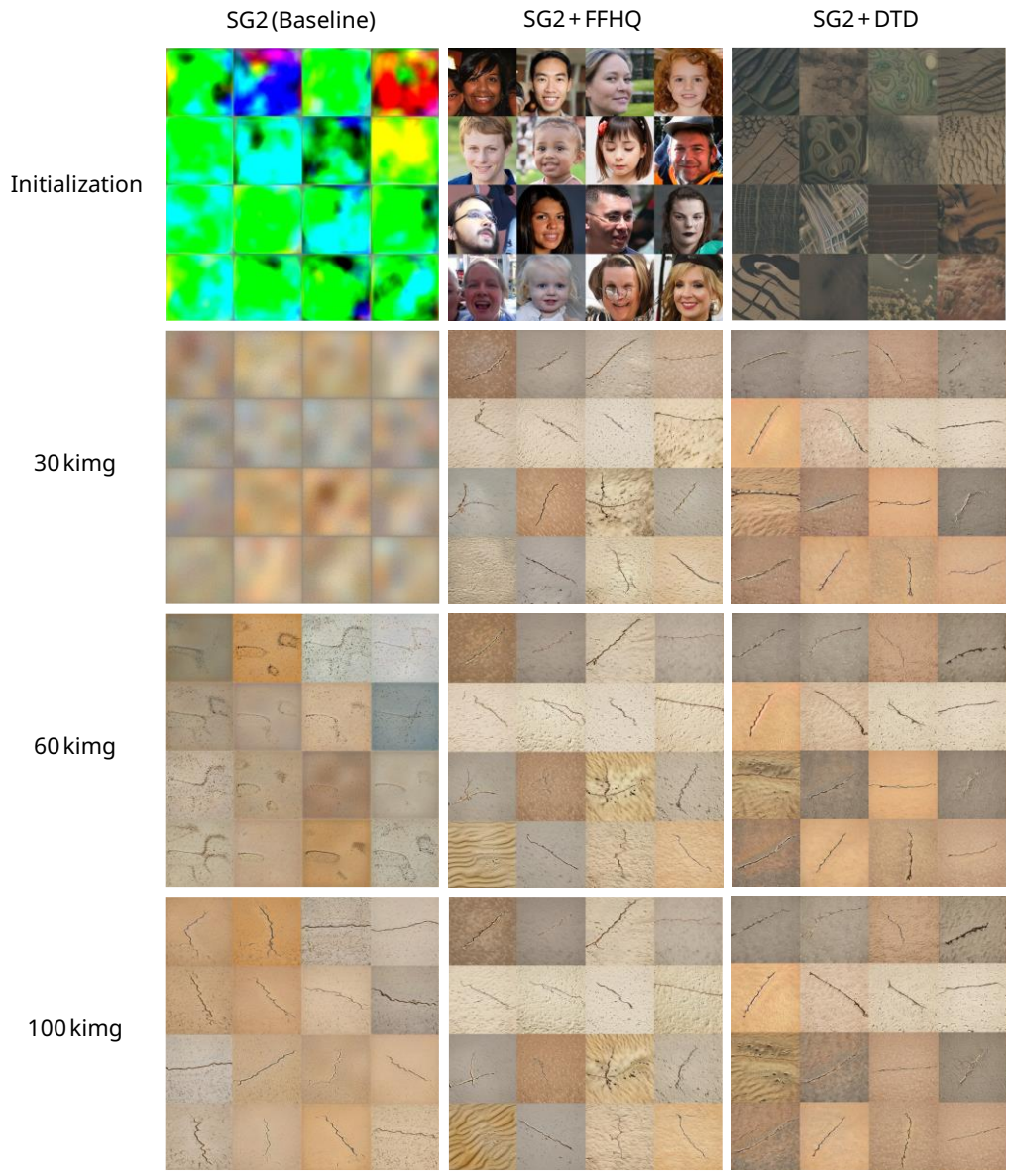


Figure 9. Early synthesis progression of the three StyleGAN2-ADA training configurations. Samples are shown at initialization, 30 kimg, 60 kimg, and 100 kimg. At initialization, the baseline configuration produces unstructured noise, while the pre-trained configurations generate coherent textures that reflect their source-domain priors. By 30 kimg, SG2 begins to acquire coarse color and texture distributions, whereas the pre-trained configurations already synthesize recognizable crack-like structures. All configurations improve in fidelity and structural realism by 60 kimg, with the pre-trained configurations presenting more developed crack morphology. By 100 kimg, all configurations produce reasonable crack patterns, although the pre-trained configurations contain sharper edges, more consistent textures, and realistic background materials, highlighting the benefits of transfer learning.

In addition to achieving the strongest FID score, SG2 + FFHQ also exhibits the highest mean (0.472 ± 0.090) and median (0.477) LPIPS, indicating that samples generated from this configuration contain greater perceptual diversity than the other StyleGAN2-ADA training configurations. The baseline configuration has a mean LPIPS of 0.452 ± 0.094, indicating moderate perceptual diversity with relatively symmetric distribution. SG2 + DTD achieves a slightly higher mean LPIPS at 0.457 ± 0.092, confirming that pre-training provides benefits for generative diversity. Interestingly, the lower bound (0.011) of the LPIPS range for SG2 indicates that the baseline configuration contains a small number of visually homogenous samples, however, the upper bound (0.725) suggests that it also

 doi:10.20944/preprints202603.0954.v1

contains a variety of meaningfully diverse surface crack images. Both pre-trained configurations exhibit higher minimum LPIPS values than SG2, suggesting that the structural priors provided by transfer learning enrich generation variation across all ranges. Compared to the UE5O dataset, SG2 achieves a 2.80 % increase in perceptual diversity, reflecting the increased variability in crack morphology afforded by generative modelling. Both SG2 + FFHQ and SG2 + DTD provide more impactful improvements to image diversity with gains of 7.49 % and 3.96 % on UE5O, respectively.

Figure 10 visualizes the t-SNE embeddings of the UE5O dataset and the three StyleGAN2-ADA training configurations, offering further insight into the diversity of the generated samples beyond LPIPS. Figure 10(a) summarizes the relationship between all datasets, depicting the broader impact of generative modelling on generation diversity. UE5O forms a series of distinct, compact clusters to the left and uppermost regions of the plot, suggesting that images from this dataset fall into discrete categories with low continuous diversity. All StyleGAN2-ADA configurations, conversely, exhibit more continuous spread with heavy overlap, indicating that the generative models learn continuous manifolds that encompass the original training distribution and extend well beyond it. While all three training configurations exhibit much wider horizontal and vertical expansion than UE5O, the denser feature embeddings seen in Figure 10(c) confirm that FFHQ pre-training injects additional structural priors not seen in SG2 and SG2 + DTD. Furthermore, this enhanced distributional overlap demonstrates the capacity of effective pre-training to bridge the domain gap and create smoother transitions between the training dataset and synthesized images. Overall, these results confirm that images generated by StyleGAN2-ADA cover a significantly broader feature space than deterministic UE5 renderings, validating the LPIPS behavior documented in Table 7 and confirming that the proposed framework enhances the diversity of training data.

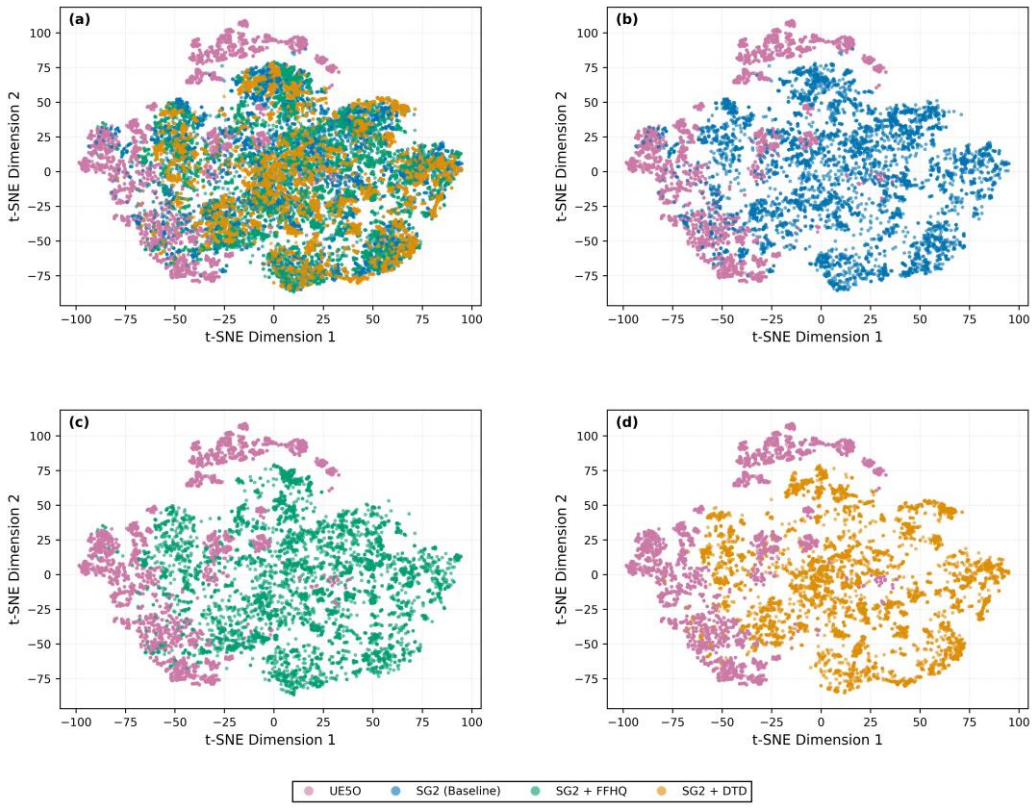


Figure 10. t-SNE visualizations of UE5O and StyleGAN2-ADA feature embeddings. (**a**) All datasets plotted together, showing UE5O forming a series of compact clusters while StyleGAN2-ADA samples occupy a broader manifold. (**b**) UE5O vs SG2 illustrates the expansion in diversity introduced by generative modelling. (**c**) UE5O vs SG2 + FFHQ highlights the large dispersion and heavy distributional overlap achieved through effective pre-training. (**d**) UE5O vs SG2 + DTD shows a similar but slightly less pronounced expansion in feature space.

 doi:10.20944/preprints202603.0954.v1

4.1.3. Qualitative Evaluation of Generation Fidelity and Diversity

To validate the trends indicated by both FID and LPIPS in the prior section, as well as explore sample-level generative behavior, we perform a qualitative evaluation on a set of representative images (examples shown in Figure 11) generated from the best checkpoints for each model.

The most observable difference between the baseline UE5O images and the StyleGAN2-ADA samples is the increased variation in crack morphology, including differences in thickness, curvature, and branching, as demonstrated in Figure 11. Specifically, Figure 11(b-c) reflects the progressive enhancement of sample diversity, with each model exhibiting additional crack propagation and illustrating new geometric structures not present in the UE5 training dataset. These visual patterns also closely mirror the quantitative results: SG2, which achieved the weakest FID and LPIPS scores, exhibits reduced structural coherence and increased texture smoothing relative to UE5O, with fewer atypical crack-like structures than the other SG2 configurations. SG2 + DTD improves sample diversity; however, generated images suffer from semantic noise, resulting in the formation of more stochastic crack patterns. SG2 + FFHQ, consistent with its highest fidelity and diversity scores, produces the most convincing samples, with sharper crack boundaries, coherent shading, and stable surface geometry. Although the UE5O baseline remains the most photorealistic, its limited morphological variation is clearly visible, aligning with the diversity constraints reflected in the LPIPS comparison. Overall, Figure 11 provides clear visual confirmation of the fidelity and diversity hierarchy quantified in Table 7, with FFHQ pre-training offering the strongest generative advantages.

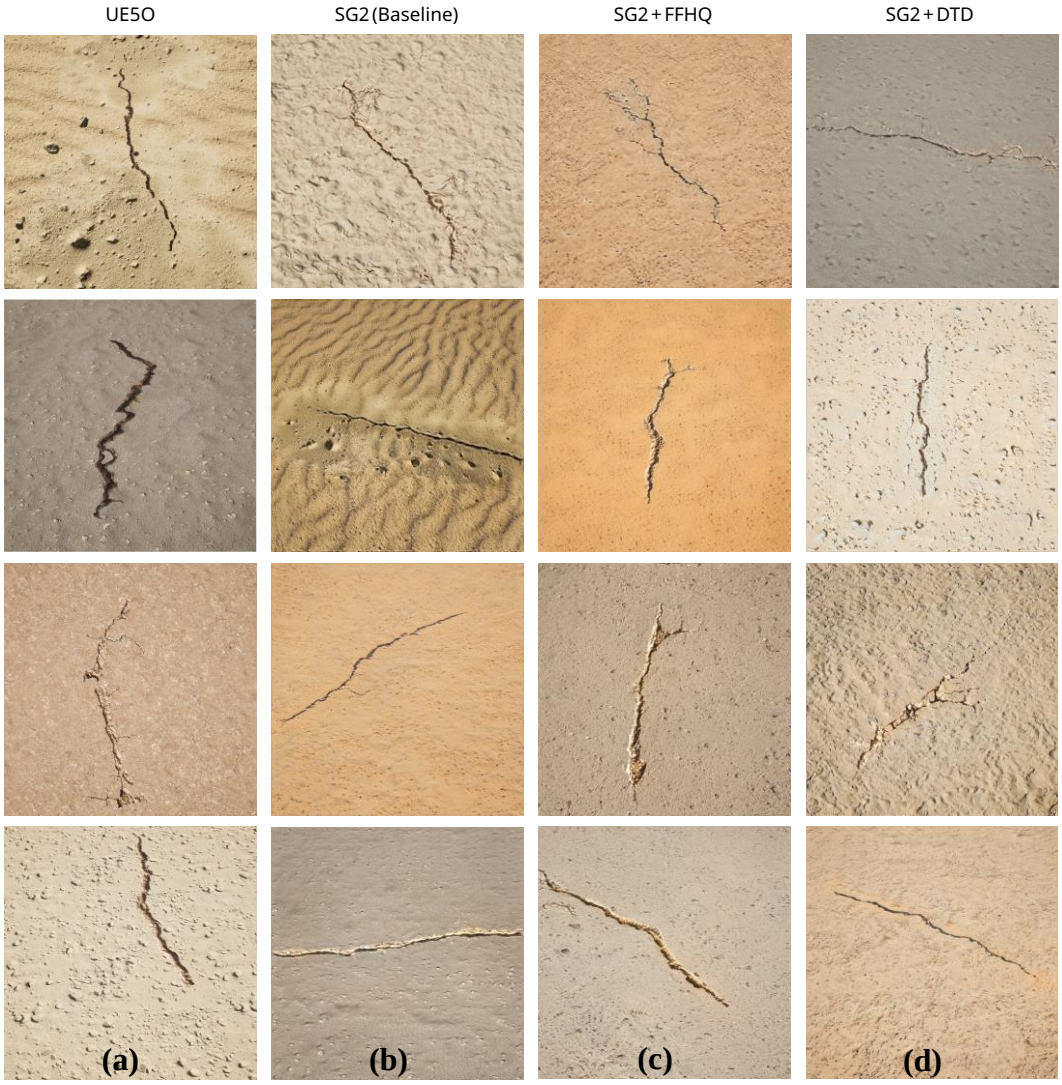


Figure 11. Example of images generated by each StyleGAN2-ADA training configuration alongside the UE5O baseline, illustrating differences in crack morphology, background texture, and overall synthesis quality

 doi:10.20944/preprints202603.0954.v1

achieved by (**a**) UE5O, (**b**) SG2 (Baseline), (**c**) SG2 + FFHQ, and (**d**) SG2 + DTD. All SG2-adapted variants, shown in subplots (**b-d**) exhibit varying degrees of improvement to overall crack morphology variation, while the baseline synthetic dataset shown in subplot (**a**) maintains greater structural coherence and photorealism.

4.2. YOLOv11 Training and Crack Detection Performance Evaluation

4.2.1. Training Dynamics

Figure 12 summarizes the YOLOv11 training behavior of the four dataset configurations developed in this study. As shown by the validation box loss curves in Figure 12(a–d), all models converged stably with no evidence of overfitting or divergence. UE5O, trained solely on game engine data, exhibited a more gradual convergence across 300 epochs, reaching a final validation box loss of 0.17. In contrast, the three SG2-adapted variants stabilized within the first 20 epochs at approximately 0.30, with training terminating at 100 epochs due to early stopping. This accelerated convergence reflects the effect of fine tuning, as the SG2-adapted models begin from weights already optimized on the UE5O dataset. All configurations achieved strong validation performance across AP@0.5, AP@[0.5:0.95], and precision, indicating that YOLOv11 learns the UE5 domain almost perfectly. This also suggests that the synthetic validation set may lack sufficient complexity to meaningfully differentiate the SG2-adapted models from the UE5O baseline.

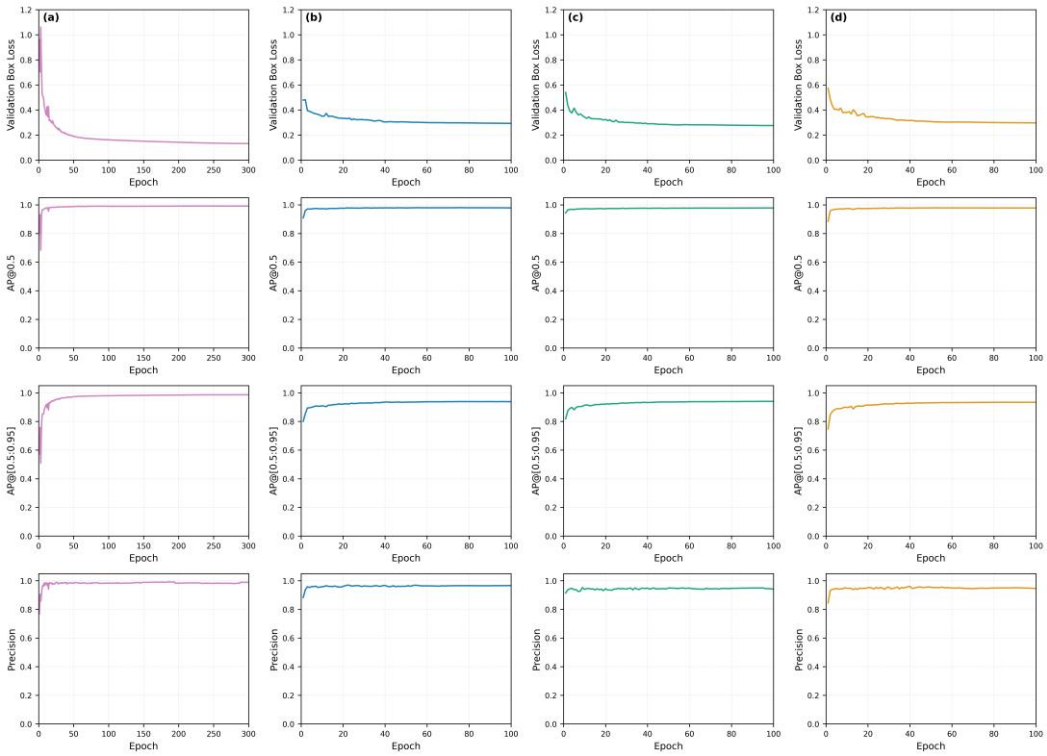


Figure 12. Training behavior of the four YOLOv11 dataset configurations evaluated in this study, showing validation box loss, AP@0.5, AP@[0.5:0.95], and precision for (**a**) UE5O, (**b**) UE5O + SG2, (**c**) UE5O + SG2-FFHQ, and (**d**) UE5O + SG2-DTD.

4.2.2. Real-World Performance Evaluation

Table 8 documents the object detection performance of the four YOLOv11 dataset configurations evaluated on a held-out test set of 200 real-world open-pit surface crack images. The UE5O baseline achieved the weakest performance overall, with an AP@0.5 of 0.403 and an AP@[0.5:0.95] of 0.223. In addition, the comparatively low precision, recall, and F1 score of this configuration highlights a clear domain gap which suggests that purely synthetic training data lacks sufficient appearance similarity to generalize effectively to real-world mining imagery. This limitation causes the model to miss more

 doi:10.20944/preprints202603.0954.v1

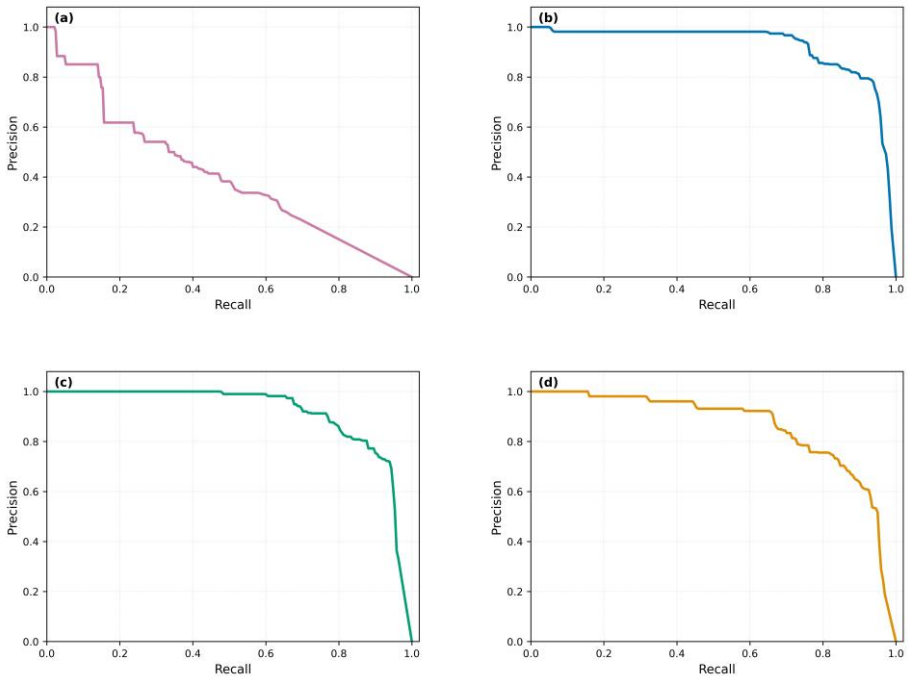
complex detections or incorrectly identify the background material as a surface crack, as illustrated in Figure 13

Table 8. YOLOv11 performance evaluation results for each dataset configuration evaluated on the real-world open-pit surface crack test set.

Dataset	Precision	Recall	F1	AP@0.5	AP@[0.5:0.95]
UE5O	0.402	0.445	0.422	0.403	0.223
UE5O + SG2	0.792	**0.902**	**0.844**	**0.922**	0.706
UE5O + SG2-FFHQ	**0.808**	0.850	0.829	0.911	**0.722**
UE5O + SG2-DTD	0.730	0.828	0.776	0.858	0.638

The precision-recall characteristics shown in Figure 14(a) further reinforce this behavior, with UE5O exhibiting a rapid monotonic degradation in precision as recall increases, indicating limited robustness to confidence threshold variation and a tendency to trade FPs for marginal gains in recall. This behavior also suggests that the real-world visual characteristics of the test set largely fall outside of the feature space of the synthetic UE5O dataset, resulting in the model struggling to distinguish between crack-like structures and background textures, leading to a disproportionately high number of FPs.

Figure 13. UE5O surface crack detection performance on real-world open-pit test images, showing low-confidence and missed detections across subplots (**a-d**). Fragmented and inconsistent spatial coverage is observed in subplots (**a**) and (**c**), while subplot (**d**) highlights a false positive triggered by background debris and shadowing effects..



 doi:10.20944/preprints202603.0954.v1

Figure 14. Precision-recall characteristics for the four YOLOv11 dataset configurations evaluated on the real-world open-pit surface crack test set. The curves demonstrate the trade-off between precision and recall across confidence thresholds, for (**a**) UE5O, (**b**) UE5O + SG2, (**c**) UE5O + SG2-FFHQ, and (**d**) UE5O + SG2-DTD, highlighting the improved robustness and extended high-precision regions of the SG2-adapted variants relative to the UE5O baseline.

This is particularly evident in Figure 13(d), where background debris is misclassified as a surface crack leading to a drop in precision. As discussed throughout Section 2.1, this behavior is consistent with prior literature, particularly for a complex domain like open-pit surface crack detection and further reinforces the need for data augmentation or domain adaptation to improve model generalizability.

All SG2-adapted variants, in contrast, demonstrate substantially improved detection performance across all reported metrics, suggesting that GAN-based refinement introduces beneficial noise and structural diversity which benefits downstream CV models. As shown in Table 8, these configurations achieved markedly higher AP, precision, recall, and F1 scores relative to the UE5O baseline, confirming that the images generated by the proposed framework provide a more effective training distribution for real-world open-pit surface crack detection. UE5O + SG2-FFHQ attained the highest precision of 0.808, indicating strong suppression of FPs, while UE5O + SG2 achieved the highest recall of 0.902, correctly identifying the largest proportion of surface cracks. Although UE5O + SG2-DTD exhibited slightly weaker performance than the other SG2 variants, it still delivered an improvement of 0.354 in F1 score over the UE5O baseline, highlighting the benefits afforded by GAN-augmented data on model generalizability.

Despite having a slightly weaker AP@0.5 than UE5O + SG2, UE5O + SG2-FFHQ provided superior localization performance across a broader range of IoU thresholds, with an AP@[0.5:0.95] of 0.722. This behavior is consistent with the generative metrics reported in Table 7, where the FFHQ-initialized StyleGAN2-ADA model achieved the highest quantitative fidelity. This suggests that the improved perceptual quality and structural coherence captured by the FFHQ pre-trained model (see Section 4.1.3) may translate into more realistic crack morphology and boundary definition, improving bounding box stability at stricter IoU thresholds and enhancing model robustness. Additionally, the improvement in perceptual diversity provided by FFHQ pre-training over SG2 baseline (see Section 4.1.2) likely enables UE5O + SG2-FFHQ to generalize more effectively to more atypical crack structures present in the real-world test set, resulting in more accurate predictions with tighter localized boundaries.

Interestingly, UE5O + SG2 achieved better recall than UE5O + SG2-FFHQ, suggesting that the baseline synthetic distribution, while exhibiting lower perceptual diversity and quantitative fidelity, may contain more pronounced crack features that allow the detector to maintain higher sensitivity, albeit at the expense of localization precision. In contrast, the comparatively weaker performance of UE5O + SG2-DTD suggests that texture-focused initialization likely emphasizes less relevant features, where, despite exhibiting competitive FID and LPIPS scores, generated samples appear to lack sufficient structural coherence for the detector to learn stable crack boundaries, leading to reduced accuracy overall. These initialization-dependent effects are additionally demonstrated by the precision-recall behavior in Figure 14(b-d), where all GAN-adapted variants consistently maintain high precision with increasing levels of recall, indicating improved robustness to confidence threshold variation relative to UE5O. For instance, UE5O + SG2 maintained near perfect precision until reaching a recall of approximately 0.70, while both UE5O + SG2-FFHQ and UE5O + SG2-DTD exhibited similarly strong performance with slightly earlier onset of precision decay at higher levels of recall.

Figure 15 further clarifies how StyleGAN2-ADA initialization influences detection confidence and localization behavior by providing a comparison of representative test detections for the four YOLOv11 dataset configurations. As shown in Figure 15(a-d), UE5O + SG2 generally produces the highest confidence predictions but occasionally includes additional background material within

 doi:10.20944/preprints202603.0954.v1

predicted bounding boxes. This suggests that the detector exhibits a tendency toward maintaining objectness confidence across threshold variations, increasing bounding box spatial coverage to maintain sufficient overlap with ground truth annotations. This qualitative behavior is consistent with the strong recall and AP@0.5 seen for UE5O + SG2 in Table 8, as well as the consistently high precision maintained with increasing levels of recall in Figure 14(b), and works to explain the slightly weaker AP@[0.5:0.95] that results from reduced localization accuracy at stricter IoU thresholds.

In contrast, UE5O + SG2-FFHQ provides more selective detections with improved spatial alignment relative to ground truth annotations, as seen by the tighter bounding boxes in Figure 15(b-c). While this behavior can result in lower confidence predictions, the increased localization accuracy leads to stronger performance at higher IoU thresholds, resulting in a stronger AP@[0.5:0.95], as seen in Table 8. UE5O + SG2-DTD attains comparatively weaker qualitative results, exhibiting reduced robustness which manifests as missed detections for lower-contrast or partially occluded cracks, such as in Figure 15(c), and FPs with overlapping predictions for background material, as observed in Figure 15(d). This behavior impacts both precision and AP, resulting in the early precision drops in Figure 14(d), and weaker overall model performance relative to the other SG2-adapted variants.

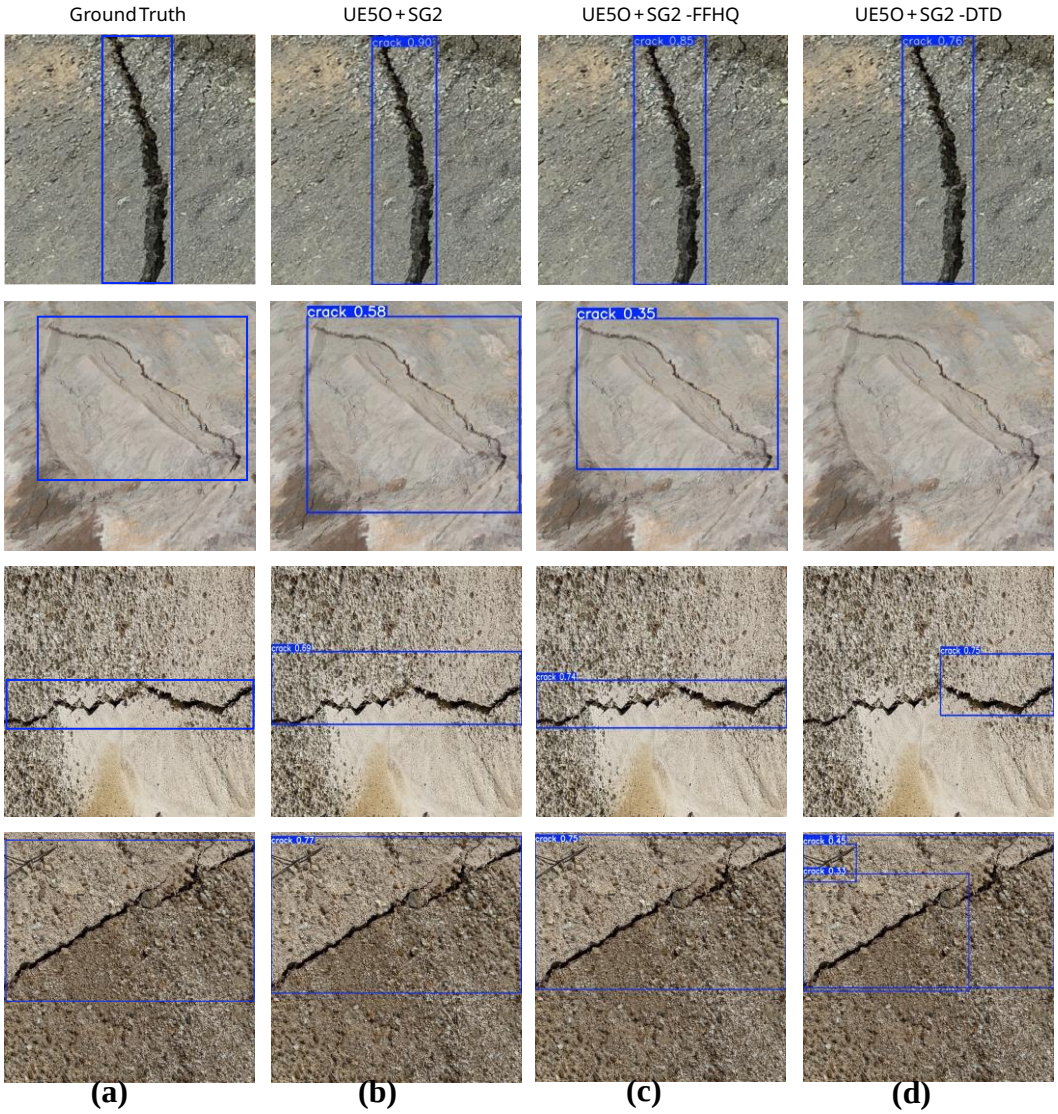


Figure 15. Qualitative comparison of SG2-adapted surface crack detection performance on real-world open-pit test images. Each row shows a test image with ground truth annotation, followed by predictions from UE5O + SG2, UE5O + SG2-FFHQ, and UE5O + SG2-DTD. Across the representative examples, UE5O + SG2 demonstrates high prediction confidence but produces less tight bounding box localization, while UE5O + SG2-FFHQ frequently provides improved spatial alignment at the cost of slightly reduced confidence (subplots **b-c**). UE5O + SG2-DTD demonstrates reduced robustness under challenging conditions, including missed detections on low

 doi:10.20944/preprints202603.0954.v1

contrast or partially occluded cracks (subplots **b-c**), as well as false positives on background debris and rubble (subplot **d**). This behavior explains the observed differences in precision–recall characteristics (Figure 15) and overall model performance (Table 8).

To further examine this behavior, Table 9 reports the distribution of FP and FN counts for each YOLOv11 dataset configuration evaluated on the real-world test set, providing a breakdown of the underlying causes for the aforementioned quantitative and qualitative performance trends. Consistent with its weaker overall performance, UE5O exhibits a comparatively high number of FPs and FNs, further quantifying the sample-level behavior seen in Figure 13. This directly contributes to the precision-recall characteristics observed in Figure 14(a), where the high FP count causes rapid precision degradation with increasing recall, with the curve terminating early due to a large number of FNs.

Table 9. FP and FN count for each YOLOv11 dataset configuration evaluated on the real-world open-pit surface crack test set.

Configuration	FP Count	FN Count
UE5O	52	95
UE5O + SG2	**27**	20
UE5O + SG2-FFHQ	42	**8**
UE5O + SG2-DTD	48	31

This domain gap is effectively closed by generative modelling, with all SG2-adapted variants demonstrating substantially reduced error rates relative to the UE5O baseline. UE5O + SG2 records the lowest FP count overall, achieving a balance between FPs and FNs consistent with its strong recall and AP@0.5 performance. The slightly weaker AP@[0.5:0.95] observed for UE5O + SG2 is further explained by Table 9, where localization imprecision causes otherwise correct detections to fail to match the ground truth annotation at stricter IoU thresholds, thereby causing an increase in the FN count. Although UE5O + SG2-FFHQ exhibits a moderately higher FP count than UE5O + SG2, the substantial reduction in missed detections at stricter IoU thresholds achieved by this configuration drives its superior localization robustness. This behavior aligns with the increased precision and AP@[0.5:0.95] seen in Table 8, manifesting in the more spatially accurate detections shown in Figure 15(b-c). UE5O + SG2-DTD demonstrates the weakest performance amongst the SG2-adapted variants, exhibiting elevated FP and FN counts relative to both UE5O + SG2 and UE5O + SG2-FFHQ consistent with the earlier precision degradation seen in Figure 14(d). As shown qualitatively in Figure 15(d), the increase in FP count for this configuration can be attributed to frequent misclassifications where background debris or rubble are incorrectly identified as surface cracks. Conversely, Figure 15(b) highlights a missed detection, suggesting that lower contrast surface cracks may be responsible for the elevated FN count seen for UE5O + SG2-DTD in Table 9.

One persistent limitation seen across all configurations, regardless of initialization strategy, is the difficulty in detecting heavily occluded, fine-grained surface cracks. This behavior, highlighted in Figure 16, causes a missed detection which impacts overall recall performance, and is likely attributable in part to the inherent architectural constraints of CNN-based object detection models such as YOLOv11 [131]. A number of factors such as backbone downsampling, localization sensitivity, and receptive field size mismatch can result in the model ignoring or missing smaller features [132]. As a result, while more pronounced cracks are generally detected with high confidence, hairline cracks hidden within the background terrain remain challenging to localize consistently.

 doi:10.20944/preprints202603.0954.v1

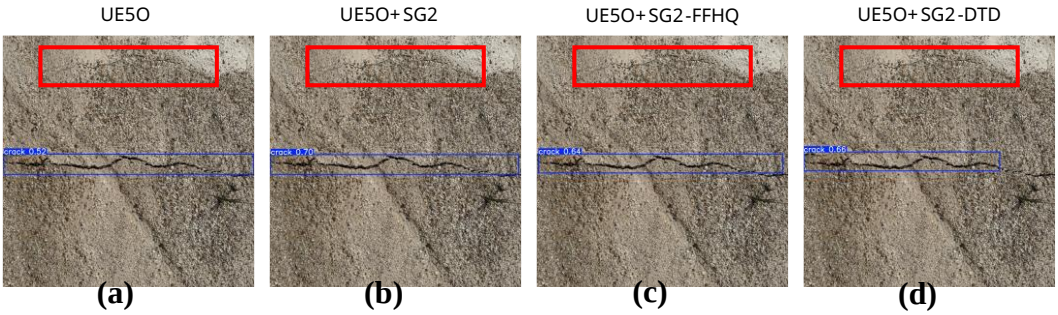


Figure 16. Representative failure case highlighting the difficulty in detecting fine-grained surface cracks in real-world open-pit test images for (**a**) UE5O, (**b**) UE5O + SG2, (**c**) UE5O + SG2-FFHQ, and (**d**) UE5O + SG2-DTD, where the red bounding box denotes the ground truth annotation for the missed detection.

4.3. Practical Deployment Considerations

While the results presented throughout the preceding sections demonstrate that the proposed hybrid synthetic dataset generation framework significantly improves real-world open-pit surface crack detection performance and alleviates the issue of data scarcity, its practical utility depends not only on detection accuracy but also on its deployability in operational settings.

A significant practical advantage of the proposed framework is its ability to generate extensive datasets at scale with minimal human intervention. Unlike conventional CV-based monitoring, which necessitates costly and laborious field data collection and manual annotation [46–48], the images synthesized in this study are labelled automatically. For mining operations, this eliminates the need to deploy workers into the field solely for data collection purposes, reducing operational disruption and safety risks. Furthermore, this decoupling of dataset size from annotation effort enables rapid dataset expansion without a proportional increase in manual labelling time, enhancing operational efficiency. Beyond this, the nature of virtualized game engine environments allows for iterative refinement and adaptation to new contexts, enabling continuous improvement of training data as domain requirements evolve. This reduces the overall cost associated with the implementation of object detection models and enhances the accessibility of domain-specific training data.

One practical consideration is the upfront computational overhead associated with both UE5-based image synthesis and StyleGAN2-ADA-based generative refinement. In this study, full dataset generation of 20 000 UE5 images required approximately six hours on high-end consumer hardware (NVIDIA GeForce RTX 5090 GPU and 64 GB RAM). Similarly, the use of StyleGAN2-ADA incurred additional GPU costs, requiring nearly 12 hours for training and a further four hours for image synthesis and pseudo-labelling per configuration on an NVIDIA Tesla A100 GPU, amounting to a total of 64 hours of GPU runtime. As such, the overall framework is best viewed as an upfront investment to generate large-scale datasets for data-scarce domains, rather than as a lightweight data augmentation technique.

Ultimately, the primary practical outcome of the proposed framework is improved downstream model robustness to real-world variability, particularly in data-scarce environments. As demonstrated in Section 4.2.2, the combination of game engine data and GAN-based generative refinement enables CV models to generalize more effectively across diverse viewpoints, crack morphologies, and background conditions, all without requiring access to real-world training datasets. Improvements in metrics essential for automated inspection workflows, such as recall and AP@[0.5:0.95], indicate that the framework trains consistent and reliable object detection models that can enhance downstream analysis and decision support. More broadly, these findings demonstrate that the absence of large-scale real-world datasets need not restrain the effective application of CV models in data-scarce domains. By carefully constructing domain-specific virtualized environments within a game engine and subsequently enriching their appearance diversity through generative modeling, it is possible to produce training data that generalize effectively to real-world conditions.

 doi:10.20944/preprints202603.0954.v1

In this context, the proposed hybrid synthetic dataset generation framework provides a practical pathway for addressing data scarcity in domains constrained by privacy, intellectual property, and proprietary concerns. For mining operations in particular, where site-specific geotechnical data is often commercially sensitive and difficult to share across organizations, this capability offers a pathway to develop robust inspection systems without compromising data security.

5. Conclusions and Future Work

Autonomous surface crack detection in open-pit mining offers numerous benefits such as enhanced worker safety and improved operational efficiency. However, CV models require large amounts of representative training data to generalize effectively to unseen conditions, impacting their applicability in commercial domains constrained by safety, cost, and data confidentiality considerations. To address this challenge, this study presented a hybrid game engine—generative AI framework for dataset synthesis and evaluated its effectiveness for surface crack detection in real-world open-pit mining imagery. The proposed approach combined the realism of the UE5 game engine with the scalability of StyleGAN2-ADA, enabling the synthesis of large-scale, fully labelled surface crack datasets that significantly improve the generalizability of CV models without reliance on extensive field data collection or manual annotation.

Comprehensive evaluation on a held-out real-world test set demonstrated that object detection models trained on images generated by the proposed framework substantially outperformed those trained solely on synthetic data from UE5. In particular, AP@0.5 increased from 0.403 to 0.922 for the best-performing GAN-adapted configuration, while AP@[0.5:0.95] exhibited approximately a threefold improvement across the board, indicating significantly enhanced localization robustness and bounding box accuracy. These performance gains were accompanied by higher recall and reduced missed detections, confirming that the proposed framework effectively narrows the domain gap between synthetic and real-world imagery through the increased diversity and realism of its generated samples. From a practical perspective, this work highlights the viability of synthetic data in autonomous inspection workflows, reducing dependence on manual labeling while mitigating operational, safety, and confidentiality constraints associated with real-world data collection. More broadly, it demonstrates that synthetic data can generalize effectively to real-world conditions when underpinned by an appropriate generation framework, suggesting that the proposed approach may be extended to other data-constrained domains where large-scale labelled datasets are similarly difficult to obtain.

Future work will iterate on this research in several ways. Firstly, diffusion-based generative models will be explored as an alternative to the proposed framework to examine whether their enhanced synthesis fidelity provides meaningful downstream benefits over the controllability of game engine-based rendering. Secondly, the detection pipeline will be extended toward multi-scale learning for improved object detection at varying distances, and instance or semantic segmentation to enable more precise delineation of surface crack boundaries for downstream analysis such as propagation measurement. Additionally, to target the limitation of small object detection identified in the study, future work will investigate architectural enhancements to the object detection and segmentation models to further improve sensitivity to fine-grained cracks. Finally, framework integration with edge-based inference platforms and UAV-based data acquisition will be examined to support real-time autonomous inspection workflows across diverse open-pit mining environments.

Author Contributions: Conceptualization, R.L., S.K., M.S., I.M.; methodology, R.L.; software, R.L.; validation, R.L.; formal analysis, R.L.; investigation, R.L.; resources, R.L.; data curation, R.L.; writing—original draft preparation, R.L.; writing—review and editing, R.L.; visualization, R.L.; supervision, S.K., M.S., I.M.; project administration, S.K., M.S., I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

 doi:10.20944/preprints202603.0954.v1

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADA	Adaptive Discriminator Augmentation
AI	Artificial Intelligence
AP	Average Precision
BERT	Bidirectional Encoder Representations from Transformers
CBS	Convolutional Block with Batch normalization and SiLU
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DL	Deep Learning
DSLR	Digital Single-Lens Reflex
DTD	Describable Textures Dataset
FC	Fully Connected
FID	Fréchet Inception Distance
FN	False Negative
FOV	Field of View
FP	False Positive
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GTA V	Grand Theft Auto V
HRSST	High Resolution Screenshot Tool
IoU	Intersection over Union
LPIPS	Learned Perceptual Image Patch Similarity
mAP	Mean Average Precision
MLP	Multilayer Perceptron
NDDS	NVIDIA Deep Learning Dataset Synthesizer
PCK	Percentage of Correct Keypoints
PBR	Physically Based Rendering
RAM	Random Access Memory
RT	Ray Tracing
RTGI	Real-Time Global Illumination
SG2	StyleGAN2-ADA
SOTA	State of the Art
SPPF	Spatial Pyramid Pooling-Fast
TAA	Temporal Anti-Aliasing
TP	True Positive
t-SNE	t-Distributed Stochastic Neighbor Embedding
UAV	Unmanned Aerial Vehicle
UE	Unreal Engine
UE4	Unreal Engine 4
UE5	Unreal Engine 5
UE5O	Unreal Engine 5 only
VAE	Variational Autoencoder
VGG-16	Visual Geometry Group-16
ViT	Vision Transformer
VLM	Vision Language Model

 doi:10.20944/preprints202603.0954.v1

References

1. Li, G.; Hu, Z.; Wang, D.; Wang, L.; Wang, Y.; Zhao, L.; Jia, H.; Fang, K. Instability Mechanisms of Slope in Open-Pit Coal Mines: From Physical and Numerical Modeling. *Int. J. Min. Sci. Technol.* **2024**, *34*, 1509–1528, doi:10.1016/j.ijmst.2024.10.003.
2. Kolapo, P.; Oniyide, G.O.; Said, K.O.; Lawal, A.I.; Onifade, M.; Munemo, P. An Overview of Slope Failure in Mining Operations. *Mining* **2022**, *2*, 350–384, doi:10.3390/mining2020019.
3. de Graaf, P.J.H.; Desjardins, M.; Tsheko, P.; Fourie, A.B.; Tibbett, M. *Geotechnical Risk Management for Open Pit Mine Closure: A Sub-Arctic and Semi-Arid Case Study*; Australian Centre for Geomechanics, 2019; pp. 211–234;.
4. Zhang, N.; Wang, Y.; Zhao, F.; Wang, T.; Zhang, K.; Fan, H.; Zhou, D.; Zhang, L.; Yan, S.; Diao, X.; et al. Monitoring and Analysis of the Collapse at Xinjing Open-Pit Mine, Inner Mongolia, China, Using Multi-Source Remote Sensing. *Remote Sens.* **2024**, *16*, 993, doi:10.3390/rs16060993.
5. Lin, Y.N.; Park, E.; Wang, Y.; Quek, Y.P.; Lim, J.; Alcantara, E.; Loc, H.H. The 2020 Hpakant Jade Mine Disaster, Myanmar: A Multi-Sensor Investigation for Slope Failure. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 291–305, doi:10.1016/j.isprsjprs.2021.05.015.
6. Martin, C.D.; Stacey, P.F.; Dight, P.M. *Pit Slopes in Weathered and Weak Rocks*; Australian Centre for Geomechanics, 2013; pp. 3–28;.
7. Zhong, Z.; Hu, B.; Li, J.; Sheng, J.; Wan, C. Impact of Rainfall Dry-Wet Cycles on Slope Deformation and Landslide Prediction in Open-Pit Mines: A Case Study of Mohuandang Landslide, Emeishan, China. *Results Eng.* **2025**, *26*, 105011, doi:10.1016/j.rineng.2025.105011.
8. Wang, W.; Griffiths, D. Case Study of Slope Failure during Construction of an Open Pit Mine in Indonesia. *Can. Geotech. J.* **2018**, *56*, 636–648, doi:10.1139/cgj-2017-0662.
9. Kong, K.W.K.; Dight, P.M. *Blasting Vibration Assessment of Rock Slopes and a Case Study*; Australian Centre for Geomechanics, 2013; pp. 1335–1344;.
10. Wang, J.; Zhou, Z.; Chen, C.; Wang, H.; Chen, Z. Failure Mechanism and Stability Analysis of an Open-Pit Slope under Excavation Unloading Conditions. *Front. Earth Sci.* **2023**, *11*, doi:10.3389/feart.2023.1109316.
11. Bridges, M.C.; Dight, P.M. *An Extensional Mechanism of Instability and Failure in the Walls of Open Pit Mines*; Australian Centre for Geomechanics, 2013; pp. 137–150;.
12. Martin, C.D.; Stacey, P.F.; Dight, P.M. *Pit Slopes in Weathered and Weak Rocks*; Australian Centre for Geomechanics, 2013; pp. 3–28;.
13. Whittall, J.R.; McDougall, S.; Eberhardt, E. A Risk-Based Methodology for Establishing Landslide Exclusion Zones in Operating Open Pit Mines. *Int. J. Rock Mech. Min. Sci.* **2017**, *100*, 100–107, doi:10.1016/j.ijrmms.2017.10.012.
14. McQuillan, A.; Canbulat, I.; Oh, J. Methods Applied in Australian Industry to Evaluate Coal Mine Slope Stability. *Int. J. Min. Sci. Technol.* **2020**, *30*, 151–155, doi:10.1016/j.ijmst.2019.11.001.
15. Vaziri, A.; Moore, L.; Ali, H. Monitoring Systems for Warning Impending Failures in Slopes and Open Pit Mines. *Nat. Hazards* **2010**, *55*, 501–512, doi:10.1007/s11069-010-9542-5.
16. Mohammed, M.M. A Review On Slope Monitoring And Application Methods In Open Pit Mining Activities. *Int. J. Min. Sci. Technol. Res.* **2021**, *10*, 181–186.
17. Ching, J.; Phoon, K.-K. Value of Geotechnical Site Investigation in Reliability-Based Design Advances in Structural Engineering. *Adv. Struct. Eng.* **2012**, *15*, 1935–1945, doi:10.1260/1369-4332.15.11.1935.
18. Zumrawi, M. Effects of Inadequate Geotechnical Investigations on Civil Engineering projects. *Int. J. Sci. Res. IJSR* **2014**, *3*, 927–931.
19. Crisp, M.P.; Jaksa, M.; Kuo, Y. Optimal Testing Locations in Geotechnical Site Investigations through the Application of a Genetic Algorithm. *Geosciences* **2020**, *10*, 265, doi:10.3390/geosciences10070265.
20. Le Roux, R.; Sepehri, M.; Khaksar, S.; Murray, I. Slope Stability Monitoring Methods and Technologies for Open-Pit Mining: A Systematic Review. *Mining* **2025**, *5*, 32, doi:10.3390/mining5020032.
21. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J. Big Data* **2021**, *8*, 53, doi:10.1186/s40537-021-00444-8.

22. Matsuzaka, Y.; Yashiro, R. AI-Based Computer Vision Techniques and Expert Systems. *AI* **2023**, *4*, 289–302, doi:10.3390/ai4010013.
23. Kalluri, P.R.; Agnew, W.; Cheng, M.; Owens, K.; Soldaini, L.; Birhane, A. Computer-Vision Research Powers Surveillance Technology. *Nature* **2025**, *643*, 73–79, doi:10.1038/s41586-025-08972-6.
24. Application of Convolution Neural Network for Digital Image Processing. *Eng. Int.* **2020**, *8*, 149–xxx, doi:10.18034/ei.v8i2.592.
25. Kameswari, C.; J, K.; Reddy, T.; Chinthaguntla, B.; Jagatheesaperumal, S.; Gaftandzhieva, S.; Doneva, R. An Overview of Vision Transformers for Image Processing: A Survey. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, doi:10.14569/IJACSA.2023.0140830.
26. Fawole, O.A.; Rawat, D.B. Recent Advances in 3D Object Detection for Self-Driving Vehicles: A Survey. *AI* **2024**, *5*, 1255–1285, doi:10.3390/ai5030061.
27. Bratulescu, R.-A.; Vatasoiu, R.-I.; Sucic, G.; Mitroi, S.-A.; Vochin, M.-C.; Sachian, M.-A. Object Detection in Autonomous Vehicles. In Proceedings of the 2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC); October 2022; pp. 375–380.
28. Albuquerque, C.; Henriques, R.; Castelli, M. Deep Learning-Based Object Detection Algorithms in Medical Imaging: Systematic Review. *Heliyon* **2025**, *11*, e41137, doi:10.1016/j.heliyon.2024.e41137.
29. Saraei, M.; Lalinia, M.; Lee, E.-J. Deep Learning-Based Medical Object Detection: A Survey. *IEEE Access* **2025**, *13*, 53019–53038, doi:10.1109/ACCESS.2025.3553087.
30. Malburg, L.; Rieder, M.-P.; Seiger, R.; Klein, P.; Bergmann, R. Object Detection for Smart Factory Processes by Machine Learning. *Procedia Comput. Sci.* **2021**, *184*, 581–588, doi:10.1016/j.procs.2021.04.009.
31. Fatima, Z.; Zardari, S.; Tanveer, M.H. Advancing Industrial Object Detection Through Domain Adaptation: A Solution for Industry 5.0. *Actuators* **2024**, *13*, 513, doi:10.3390/act13120513.
32. Di Mucci, V.M.; Cardellicchio, A.; Ruggieri, S.; Nettis, A.; Renò, V.; Uva, G. Artificial Intelligence in Structural Health Management of Existing Bridges. *Autom. Constr.* **2024**, *167*, 105719, doi:10.1016/j.autcon.2024.105719.
33. Plevris, V.; Papazafeiropoulos, G. AI in Structural Health Monitoring for Infrastructure Maintenance and Safety. *Infrastructures* **2024**, *9*, 225, doi:10.3390/infrastructures9120225.
34. Lee, J.; Lee, S. Construction Site Safety Management: A Computer Vision and Deep Learning Approach. *Sensors* **2023**, *23*, 944, doi:10.3390/s23020944.
35. Rabbi, A.B.K.; Jeelani, I. AI Integration in Construction Safety: Current State, Challenges, and Future Opportunities in Text, Vision, and Audio Based Applications. *Autom. Constr.* **2024**, *164*, 105443, doi:10.1016/j.autcon.2024.105443.
36. Ruan, S.; Hu, Y.; Liu, J.; Wang, J. An Advanced Crack Detection Method for Slope Management in Open-Pit Mines: Applying Enhanced YOLOv8 Network. *Int. J. Min. Reclam. Environ. 0*, 1–18, doi:10.1080/17480930.2025.2484477.
37. An, J.; Dong, S.; Wang, X.; Li, C.; Zhao, W. Research on UAV Aerial Imagery Detection Algorithm for Mining-Induced Surface Cracks Based on Improved YOLOv10. *Sci. Rep.* **2025**, *15*, 30101, doi:10.1038/s41598-025-14880-6.
38. Ruan, S.; Liu, D.; Gu, Q.; Jing, Y. An Intelligent Detection Method for Open-Pit Slope Fracture Based on the Improved Mask R-CNN. *J. Min. Sci.* **2022**, *58*, 503–518, doi:10.1134/S1062739122030176.
39. Letshwiti, T.M.; Shahsavar, M.; Moniri-Morad, A.; Sattarvand, J. Deep Learning-Based Image Segmentation for Highwall Stability Monitoring in Open Pit Mines. *J. Eng. Res.* **2025**, doi:10.1016/j.jer.2025.04.002.
40. Wang, K.; Wei, B.; Zhao, T.; Wu, G.; Zhang, J.; Zhu, L.; Wang, L. An Automated Approach for Mapping Mining-Induced Fissures Using CNNs and UAS Photogrammetry. *Remote Sens.* **2024**, *16*, 2090, doi:10.3390/rs16122090.
41. Winkelmaier, G.; Battulwar, R.; Khoshdeli, M.; Valencia, J.; Sattarvand, J.; Parvin, B. Topographically Guided UAV for Identifying Tension Cracks Using Image-Based Analytics in Open-Pit Mines. *IEEE Trans. Ind. Electron.* **2021**, *68*, 5415–5424, doi:10.1109/TIE.2020.2992011.
42. Bansal, Ms.A.; Sharma, Dr.R.; Kathuria, Dr.M. A Systematic Review on Data Scarcity Problem in Deep Learning: Solution and Applications. *ACM Comput Surv* **2022**, *54*, 208:1-208:29, doi:10.1145/3502287.

 doi:10.20944/preprints202603.0954.v1

43. Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; Yu, P.S. Generalizing to Unseen Domains: A Survey on Domain Generalization 2022.
44. Alzubaidi, L.; Bai, J.; Al-Sabaawi, A.; Santamaría, J.; Albahri, A.S.; Al-dabbagh, B.S.N.; Fadhel, M.A.; Manoufali, M.; Zhang, J.; Al-Timemy, A.H.; et al. A Survey on Deep Learning Tools Dealing with Data Scarcity: Definitions, Challenges, Solutions, Tips, and Applications. *J. Big Data* **2023**, *10*, 46, doi:10.1186/s40537-023-00727-2.
45. Harle, S.M.; Wankhade, R.L. Machine Learning Techniques for Predictive Modelling in Geotechnical Engineering: A Succinct Review. *Discov. Civ. Eng.* **2025**, *2*, 86, doi:10.1007/s44290-025-00224-w.
46. Ramasamy, D.; Sivamani, S. The Future of Geotechnical Engineering Through Deep Learning: A Concise Literature Review. *J. Inf. Syst. Eng. Manag.* **2025**, *10*, 685–694, doi:10.52783/jisem.v10i14s.2380.
47. Yamani, A.; AlAmoudi, N.; Albilali, S.; Baslyman, M.; Hassine, J. Data Requirement Goal Modeling for Machine Learning Systems 2025.
48. Taye, M.M. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers* **2023**, *12*, 91, doi:10.3390/computers12050091.
49. Hutchinson, M.L.; Antono, E.; Gibbons, B.M.; Paradiso, S.; Ling, J.; Meredig, B. Overcoming Data Scarcity with Transfer Learning 2017.
50. Wang, Z.; Wang, P.; Liu, K.; Wang, P.; Fu, Y.; Lu, C.-T.; Aggarwal, C.C.; Pei, J.; Zhou, Y. A Comprehensive Survey on Data Augmentation 2025.
51. Zhao, Z.; Alzubaidi, L.; Zhang, J.; Duan, Y.; Gu, Y. A Comparison Review of Transfer Learning and Self-Supervised Learning: Definitions, Applications, Advantages and Limitations. *Expert Syst. Appl.* **2024**, *242*, 122807, doi:10.1016/j.eswa.2023.122807.
52. Brodzicki, A.; Piekarski, M.; Kucharski, D.; Jaworek-Korjakowska, J.; Gorgon, M. Transfer Learning Methods as a New Approach in Computer Vision Tasks with Small Datasets. *Found. Comput. Decis. Sci.* **2020**, *45*, 179–193, doi:10.2478/fcds-2020-0010.
53. Kumar, T.; Mileo, A.; Brennan, R.; Bendechache, M. Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions 2023.
54. Mumuni, A.; Mumuni, F. Data Augmentation: A Comprehensive Survey of Modern Approaches. *Array* **2022**, *16*, 100258, doi:10.1016/j.array.2022.100258.
55. Li, M.; Chen, H.; Wang, Y.; Zhu, T.; Zhang, W.; Zhu, K.; Wong, K.-F.; Wang, J. Understanding and Mitigating the Bias Inheritance in LLM-Based Data Augmentation on Downstream Tasks 2025.
56. Nikolenko, S. *Synthetic Data for Deep Learning*; 2021; ISBN 978-3-030-75177-7.
57. Unreal Engine 5 Available online: https://www.unrealengine.com/en-US/unreal-engine-5 (accessed on 15 September 2025).
58. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine Available online: https://unity.com (accessed on 15 September 2025).
59. Li, Y.; Dong, X.; Chen, C.; Li, J.; Wen, Y.; Spranger, M.; Lyu, L. Is Synthetic Image Useful for Transfer Learning? An Investigation into Data Generation, Volume, and Utilization 2024.
60. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks 2014.
61. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks 2019.
62. Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; Aila, T. Training Generative Adversarial Networks with Limited Data 2020.
63. Synthetic Scientific Image Generation with VAE, GAN, and Diffusion Model Architectures Available online: https://www.mdpi.com/2313-433X/11/8/252 (accessed on 28 November 2025).
64. Werda, M.S.; Taibi, H.; Kouiss, K.; Chebak, A.; Ben Halima, S.; Decottignies, M.; Dilliott, C. Towards Minimizing Domain Gap When Using Synthetic Data in Automotive Vision Control Applications. *IFAC-Pap.* **2024**, *58*, 522–527, doi:10.1016/j.ifacol.2024.09.265.
65. Bandi, A.; Adapa, P.V.S.R.; Kuchi, Y.E.V.P.K. The Power of Generative AI: A Review of Requirements, Models, Input–Output Formats, Evaluation Metrics, and Challenges. *Future Internet* **2023**, *15*, 260, doi:10.3390/fi15080260.

 doi:10.20944/preprints202603.0954.v1

66. Jang, Y.; Baek, J.; Jeon, S.; Han, S. Bridging the Simulation-to-Real Gap of Depth Images for Deep Reinforcement Learning. *Expert Syst. Appl.* **2024**, *253*, 124310, doi:10.1016/j.eswa.2024.124310.
67. Marvellous, A. Sim-to-Real Transfer in Robotic Manipulation Using Domain Randomization and Meta-Learning. *Robotics* **2025**.
68. Ulhas, S.S.; Kannapiran, S.; Berman, S. GAN-Based Domain Adaptation for Creating Digital Twins of Small-Scale Driving Testbeds: Opportunities and Challenges. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV); June 2024; pp. 137–143.
69. Man, K.; Chahl, J. A Review of Synthetic Image Data and Its Use in Computer Vision. *J. Imaging* **2022**, *8*, 310, doi:10.3390/jimaging8110310.
70. Half-Life 2 on Steam Available online: https://store.steampowered.com/app/220/HalfLife_2/ (accessed on 9 October 2025).
71. Taylor, G.R.; Chosak, A.J.; Brewer, P.C. OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition; June 2007; pp. 1–8.
72. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for Data: Ground Truth from Computer Games 2016.
73. Hwang, H.; Adhikari, K.; Shodhaka, S.; Kim, D. Synthetic Data Augmentation for Robotic Mobility Aids to Support Blind and Low Vision People. In Proceedings of the Robot Intelligence Technology and Applications 9; Park, D., Liu, C., Lee, D.-Y., Kim, M.J., Eds.; Springer Nature Switzerland: Cham, 2025; pp. 92–102.
74. Lee, H.; Jeon, J.; Lee, D.; Park, C.; Kim, J.; Lee, D. Game Engine-Driven Synthetic Data Generation for Computer Vision-Based Safety Monitoring of Construction Workers. *Autom. Constr.* **2023**, *155*, 105060, doi:10.1016/j.autcon.2023.105060.
75. NVIDIA/Dataset_Synthesizer 2025.
76. Perception Package | Perception Package | 1.0.0-Preview.1 Available online: https://docs.unity3d.com/Packages/com.unity.perception@1.0/manual/index.html (accessed on 9 October 2025).
77. Games, R. Grand Theft Auto V Available online: https://www.rockstargames.com/gta-v (accessed on 10 October 2025).
78. Angus, M.; ElBalkini, M.; Khan, S.; Harakeh, A.; Andrienko, O.; Reading, C.; Waslander, S.; Czarnecki, K. Unlimited Road-Scene Synthetic Annotation (URSA) Dataset. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC); November 2018; pp. 985–992.
79. Rasmussen, I.; Kvalsvik, S.; Andersen, P.-A.; Aune, T.N.; Hagen, D. Development of a Novel Object Detection System Based on Synthetic Data Generated from Unreal Game Engine. *Appl. Sci.* **2022**, *12*, 8534, doi:10.3390/app12178534.
80. Turkcan, M.K.; Li, Y.; Zang, C.; Ghaderi, J.; Zussman, G.; Kostic, Z. Boundless: Generating Photorealistic Synthetic Data for Object Detection in Urban Streetscapes 2024.
81. Borkman, S.; Crespi, A.; Dhakad, S.; Ganguly, S.; Hogins, J.; Jhang, Y.-C.; Kamalzadeh, M.; Li, B.; Leal, S.; Parisi, P.; et al. Unity Perception: Generate Synthetic Data for Computer Vision 2021.
82. Cauli, N.; Reforgiato Recupero, D. Synthetic Data Augmentation for Video Action Classification Using Unity. *IEEE Access* **2024**, *12*, 156172–156183, doi:10.1109/ACCESS.2024.3485199.
83. Naidoo, J.; Bates, N.; Gee, T.; Nejati, M. Pallet Detection from Synthetic Data Using Game Engines 2023.
84. Shooter, M.; Malleson, C.; Hilton, A. SyDog: A Synthetic Dog Dataset for Improved 2D Pose Estimation 2021.
85. Lee, J.G.; Hwang, J.; Chi, S.; Seo, J. Synthetic Image Dataset Development for Vision-Based Construction Equipment Detection. *J. Comput. Civ. Eng.* **2022**, *36*, 04022020, doi:10.1061/(ASCE)CP.1943-5487.0001035.
86. Natarajan, S.A.; Madden, M.G. Hybrid Synthetic Data Generation Pipeline That Outperforms Real Data. *J. Electron. Imaging* **2023**, *32*, 023011, doi:10.1117/1.JEI.32.2.023011.
87. Sengar, S.S.; Hasan, A.B.; Kumar, S.; Carroll, F. Generative Artificial Intelligence: A Systematic Review and Applications 2024.

 doi:10.20944/preprints202603.0954.v1

88. Deijn, R. de; Batra, A.; Koch, B.; Mansoor, N.; Makkena, H. Reviewing FID and SID Metrics on Generative Adversarial Networks. In Proceedings of the AI, Machine Learning and Applications; January 27 2024; pp. 111–124.
89. Wang, R.; Chen, X.; Wang, X.; Wang, H.; Qian, C.; Yao, L.; Zhang, K. A Novel Approach for Melanoma Detection Utilizing GAN Synthesis and Vision Transformer. *Comput. Biol. Med.* **2024**, *176*, 108572, doi:10.1016/j.compbiomed.2024.108572.
90. Lai, M.; Marzi, C.; Mascalchi, M.; Diciotti, S. Brain MRI Synthesis Using Stylegan2-ADA. In Proceedings of the 2024 IEEE International Symposium on Biomedical Imaging (ISBI); May 2024; pp. 1–5.
91. Gonçalves, B.; Vieira, P.; Vieira, A. Abdominal MRI Synthesis Using StyleGAN2-ADA. In Proceedings of the 2023 IST-Africa Conference (IST-Africa); May 2023; pp. 1–9.
92. Tariq, U.; Qureshi, R.; Zafar, A.; Aftab, D.; Wu, J.; Alam, T.; Shah, Z.; Ali, H. Brain Tumor Synthetic Data Generation with Adaptive StyleGANs. In Proceedings of the Artificial Intelligence and Cognitive Science; Longo, L., O'Reilly, R., Eds.; Springer Nature Switzerland: Cham, 2023; pp. 147–159.
93. Chong, M.J.; Forsyth, D. Effectively Unbiased FID and Inception Score and Where to Find Them.; 2020; pp. 6070–6079.
94. Jayasumana, S.; Ramalingam, S.; Veit, A.; Glasner, D.; Chakrabarti, A.; Kumar, S. Rethinking FID: Towards a Better Evaluation Metric for Image Generation.; 2024; pp. 9307–9315.
95. Yang, S.; Kim, K.-D.; Ariji, E.; Takata, N.; Kise, Y. Evaluating the Performance of Generative Adversarial Network-Synthesized Periapical Images in Classifying C-Shaped Root Canals. *Sci. Rep.* **2023**, *13*, 18038, doi:10.1038/s41598-023-45290-1.
96. Fedoruk, O.; Klimaszewski, K.; Ogonowski, A.; Możdżonek, R. Performance of GAN-Based Augmentation for Deep Learning COVID-19 Image Classification. *AIP Conf. Proc.* **2024**, *3061*, 030001, doi:10.1063/5.0203379.
97. Ferreira, I.; Ochoa, L.; Koeshidayatullah, A. On the Generation of Realistic Synthetic Petrographic Datasets Using a Style-Based GAN. *Sci. Rep.* **2022**, *12*, 12845, doi:10.1038/s41598-022-16034-4.
98. Ghosh, R.; Yamany, M.S.; Smadi, O. Generation of Synthetic Dataset to Improve Deep Learning Models for Pavement Distress Assessment. *Innov. Infrastruct. Solut.* **2025**, *10*, 41, doi:10.1007/s41062-024-01850-6.
99. Feng, X.; Du, J.; Wu, M.; Chai, B.; Miao, F.; Wang, Y. Potential of Synthetic Images in Landslide Segmentation in Data-Poor Scenario: A Framework Combining GAN and Transformer Models. *Landslides* **2024**, *21*, 2211–2226, doi:10.1007/s10346-024-02274-0.
100. Achicanoy, H.; Chaves, D.; Trujillo, M. StyleGANs and Transfer Learning for Generating Synthetic Images in Industrial Applications. *Symmetry* **2021**, *13*, 1497, doi:10.3390/sym13081497.
101. NVlabs/Ffhq-Dataset 2025.
102. Barrientos-Espillco, F.; Gascó, E.; López-González, C.I.; Gómez-Silva, M.J.; Pajares, G. Semantic Segmentation Based on Deep Learning for the Detection of Cyanobacterial Harmful Algal Blooms (CyanoHABs) Using Synthetic Images. *Appl. Soft Comput.* **2023**, *141*, 110315, doi:10.1016/j.asoc.2023.110315.
103. Park, G.; Lee, Y. Wildfire Smoke Detection Enhanced by Image Augmentation with StyleGAN2-ADA for YOLOv8 and RT-DETR Models. *Fire* **2024**, *7*, 369, doi:10.3390/fire7100369.
104. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection 2024.
105. Nanite Virtualized Geometry in Unreal Engine | Unreal Engine 5.6 Documentation | Epic Developer Community Available online: https://dev.epicgames.com/documentation/en-us/unreal-engine/nanite-virtualized-geometry-in-unreal-engine (accessed on 16 September 2025).
106. Lumen Global Illumination and Reflections in Unreal Engine | Unreal Engine 5.6 Documentation | Epic Developer Community Available online: https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine (accessed on 16 September 2025).
107. Creating Landscapes in Unreal Engine | Unreal Engine 5.7 Documentation | Epic Developer Community Available online: https://dev.epicgames.com/documentation/en-us/unreal-engine/creating-landscapes-in-unreal-engine (accessed on 7 January 2026).

 doi:10.20944/preprints202603.0954.v1

108. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization 2018.
109. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World 2017.
110. Viscarra Rossel, R.A.; Bui, E.N.; de Caritat, P.; McKenzie, N.J. Mapping Iron Oxides and the Color of Australian Soil Using Visible–near-Infrared Reflectance Spectra. *J. Geophys. Res. Earth Surf.* **2010**, *115*, doi:10.1029/2009JF001645.
111. van Vreeswyk, A.; Leighton, K.; Payne, A.; Hennig, P. An Inventory and Condition Survey of the Pilbara Region, Western Australia. *Tech. Bull.* **2004**.
112. Quixel Available online: https://quixel.com/megascans (accessed on 8 January 2026).
113. Xiao, Y.; Deng, H.; Li, J.; Zhou, M.; Assefa, E.; Chen, X. A Quantitative Method for the Determination of Rock Fragmentation Based on Crack Density and Crack Saturation. *Sci. Rep.* **2023**, *13*, 11747, doi:10.1038/s41598-023-38911-2.
114. Wang, X.; Wang, Y.; Wang, Y.; Chan, T.O. A Fast and Reliable Crack Measurement Approach Based on Perspective Projection Simulation Models and UAV Imaging for Dam and Levee Inspections. *Surv. Rev.* **2025**, *0*, 1–12, doi:10.1080/00396265.2025.2486713.
115. Cine Camera Actor | Unreal Engine 4.27 Documentation | Epic Developer Community Available online: https://dev.epicgames.com/documentation/en-us/unreal-engine/cinematic-cameras-in-unreal-engine (accessed on 7 January 2026).
116. Taking Screenshots in Unreal Engine | Unreal Engine 5.7 Documentation | Epic Developer Community Available online: https://dev.epicgames.com/documentation/en-us/unreal-engine/taking-screenshots-in-unreal-engine (accessed on 9 January 2026).
117. UnrealEngine/Engine/Source/Runtime/Core/Public/Math/PerspectiveMatrix.h at Ue5-Main · EpicGames/UnrealEngine Available online: https://github.com/EpicGames/UnrealEngine/blob/ue5-main/Engine/Source/Runtime/Core/Public/Math/PerspectiveMatrix.h (accessed on 13 January 2026).
118. Ultralytics Object Detection Datasets Overview Available online: https://docs.ultralytics.com/datasets/detect/ (accessed on 13 January 2026).
119. Describable Textures Dataset Available online: https://www.robots.ox.ac.uk/~vgg/data/dtd/ (accessed on 29 October 2025).
120. Pinkney, J.N.M. Awesome Pretrained StyleGAN2 2025.
121. NVlabs/Stylegan3 2025.
122. NVlabs/Stylegan2-Ada-Pytorch 2025.
123. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium 2018.
124. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision 2015.
125. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric 2018.
126. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition 2015.
127. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows 2021.
128. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding 2019.
129. Ultralytics Ultralytics YOLO11 Available online: https://docs.ultralytics.com/models/yolo11/ (accessed on 20 January 2026).
130. Nash, J. Non-Cooperative Games. *Ann. Math.* **1951**, *54*, 286–295, doi:10.2307/1969529.

 doi:10.20944/preprints202603.0954.v1

131. Liu, H.; Li, X.; Wang, L.; Zhang, Y.; Wang, Z.; Lu, Q. MS-YOLOv11: A Wavelet-Enhanced Multi-Scale Network for Small Object Detection in Remote Sensing Images. *Sensors* **2025**, *25*, doi:10.3390/s25196008.
132. Mu, D.; Guou, Y.; Wang, W.; Peng, R.; Guo, C.; Marinello, F.; Xie, Y.; Huang, Q. URT-YOLOv11: A Large Receptive Field Algorithm for Detecting Tomato Ripening Under Different Field Conditions. *Agriculture* **2025**, *15*, doi:10.3390/agriculture15101060.