Journal of
Marine Science and Engineering

MDPI

Article

Deep Learning Approaches for Fault Detection in Subsea Oil and Gas Pipelines: A Focus on Leak Detection Using Visual Data

Viviane F. da Silva *, Theodoro A. Netto and Bessie A. Ribeiro

Subsea Technology Laboratory, Federal University of Rio de Janeiro, Rio de Janeiro 21941-914, Brazil; tanetto@lts.coppe.ufrj.br (T.A.N.)
* Correspondence: vivianeferreira1002@gmail.com

Abstract

The integrity of subsea oil and gas pipelines is essential for offshore safety and environmental protection. Conventional leak detection approaches, such as manual inspection and indirect sensing, are often costly, time-consuming, and prone to subjectivity, motivating the development of automated methods. In this study, we present a deep learning-based framework for detecting underwater leaks using images acquired in controlled experiments designed to reproduce representative conditions of subsea monitoring. The dataset was generated by simulating both gas and liquid leaks in a water tank environment, under scenarios that mimic challenges observed during Remotely Operated Vehicle (ROV) inspections along the Brazilian coast. It was further complemented with artificially generated synthetic images (Stable Diffusion) and publicly available subsea imagery. Multiple Convolutional Neural Network (CNN) architectures, including VGG16, ResNet50, InceptionV3, DenseNet121, InceptionResNetV2, EfficientNetB0, and a lightweight custom CNN, were trained with transfer learning and evaluated on validation and blind test sets. The best-performing models achieved stable performance during training and validation, with macro F1-scores above 0.80, and demonstrated improved generalization compared to traditional baselines such as VGG16. In blind testing, InceptionV3 achieved the most balanced performance across the three classes when trained with synthetic data and augmentation. The study demonstrates the feasibility of applying CNNs for vision-based leak detection in complex underwater environments. A key contribution is the release of a novel experimentally generated dataset, which supports reproducibility and establishes a benchmark for advancing automated subsea inspection methods.

Keywords: subsea pipeline leak detection; underwater image-based inspection; visual fault detection; deep learning; convolutional neural networks

check for updates

Academic Editor: Bruno Brunone

Received: 5 August 2025
Revised: 24 August 2025
Accepted: 28 August 2025
Published: 1 September 2025

Citation: Silva, V.F.d.; Netto, T.A.; Ribeiro, B.A. Deep Learning Approaches for Fault Detection in Subsea Oil and Gas Pipelines: A Focus on Leak Detection Using Visual Data. *J. Mar. Sci. Eng.* **2025**, *13*, 1683. https://doi.org/10.3390/jmse13091683

1. Introduction

The integrity of subsea pipelines is essential to the safe and efficient operation of offshore oil and gas production systems. Failures in these structures can result in serious environmental damage, financial loss, and operational setbacks, including oil spills, gas leaks, and emergency shutdowns [1–3]. As a result, early fault detection in subsea infrastructure has become a strategic priority for the energy industry.

The offshore environment poses a number of challenges to the structural integrity and long-term monitoring of submerged assets. These challenges include hydrodynamic forces, wave-induced fatigue, and complex flow regimes that affect not only dynamic devices such as energy converters but also static infrastructure like pipelines. Korde and

Ertekin [4] demonstrated how submerged buoy systems respond to irregular wave fields, reinforcing the need for reliable structural design and monitoring solutions. In addition, recent optimization approaches applied to submarine structures [5] highlight how advanced numerical methods can contribute to safer and more efficient subsea equipment design. In this context, inspection technologies that enable real-time or near-real-time assessment, particularly visual methods, can offer critical support in identifying early-stage deterioration.

Pipelines laid on the seafloor are exposed to various physical interactions, especially in areas with erodible seabeds. Recent experimental work [6] has shown that pipeline configuration directly influences scour and turbulence patterns, which can in turn accelerate degradation processes. These findings highlight the importance of monitoring systems that can detect faults at an early stage, before significant damage occurs.

Conventional inspection techniques such as visual surveys via Remotely Operated Vehicles (ROVs) or Autonomous Underwater Vehicles (AUVs), along with pressure testing, are widely used but have limitations. These methods can be costly, time-consuming, and prone to human error [7]. Advances in computer vision and machine learning, particularly the use of deep learning for image classification, have opened up new possibilities for automating inspections and improving reliability [8,9]. However, much of the current research in this field depends on synthetic data or specialized sensors, which may not generalize well to real-world scenarios and often lack reproducibility.

The present study explores the application of deep learning for leak detection in subsea pipelines using underwater images captured under controlled experimental conditions. A custom dataset was created by simulating both gas and liquid leaks in a tank environment, along with non-leak scenarios, under lighting conditions representative of subsea inspections. This dataset was further complemented with artificially generated synthetic images (Stable Diffusion) to expand its variability and realism. In addition, an independent set of publicly available subsea images from the National Oceanic and Atmospheric Administration (NOAA) of the United States of America was used exclusively for blind testing, ensuring an unbiased evaluation of generalization to real-world conditions.

Pretrained models were not used as-is; instead, they were fine-tuned with this custom dataset to ensure that the learning process was tailored to the visual patterns and challenges specific to underwater leak detection. Importantly, to the best of our knowledge, no public dataset provides real underwater leak imagery obtained under controlled experimental conditions. While public collections such as NOAA repositories include examples of field events (e.g., oil spills or methane seeps), these images are heterogeneous and were used here exclusively for blind testing, rather than for model training or validation. This dataset has now been made publicly available to support reproducibility and benchmarking in future studies.

Multiple Convolutional Neural Network (CNN) architectures were tested in this study, including VGG16, ResNet50, InceptionV3, DenseNet121, InceptionResNetV2, and EfficientNetB0, as well as a lightweight custom model. All were trained using transfer learning, starting from pretrained weights and adapting the models to the underwater leak detection task. The results demonstrate that it is possible to classify underwater imagery into three categories: non-leak, liquid leak, and gas leak, using only standard RGB images, without requiring specialized sensors. This approach offers a potentially scalable solution for integration into ROV-based inspection workflows or embedded monitoring systems.

This work addresses key gaps in the literature by providing a novel dataset that combines experimentally observed leaks with synthetic imagery for training and validation, and evaluates a public NOAA dataset exclusively for blind testing. This design enables reproducible evaluation in visually realistic conditions and a fair assessment of model generalization to unseen real-world data. The dataset and methodological framework presented here offer a reproducible reference for future research, addressing the current scarcity of publicly available underwater fault imagery. In addition, the systematic comparison across four distinct training configurations (no synthetic/no augmentation, synthetic only, augmentation only, and synthetic plus augmentation) provides a structured way to isolate and quantify the contribution of each data enrichment strategy, which has not been explicitly examined in prior studies for this particular application. Finally, by analyzing performance across multiple CNN architectures, including a lightweight custom model, this study offers insight into trade-offs between accuracy and computational efficiency, which is critical for real-time deployment on ROVs or embedded platforms operating under hardware constraints.

The rest of the paper is organized as follows: Section 2 reviews related work in signal-based and image-based detection; Section 3 details the experimental setup, dataset composition, and modeling approach; Section 4 presents and analyzes the results of multi-class classification across several CNN backbones; and Section 5 concludes the paper with final considerations and directions for future research.

2. Related Work

Recent advances in deep learning have opened relevant paths for automating the inspection of underwater structures, including pipelines. Most existing studies can be grouped into three main categories: signal-based approaches, synthetic visual data analysis, and image-based computer vision models. Each presents important contributions, but also limitations.

In parallel, traditional mathematical and hydraulic models have long been proposed as pipeline leak detection systems. However, building a model that can describe the pipeline under all operational conditions is neither simple nor accurate due to the large number of variables and empirical formulas involved, which often compromise cost-effectiveness [10]. In this context, transient test-based techniques have emerged as a relevant approach, relying on hydraulic pressure wave analysis to infer the occurrence and location of leaks. Recent advances by Meniconi et al. [11,12] have demonstrated the feasibility of applying transient tests to the Trieste subsea pipeline, reinforcing their potential for practical field deployment. While such methods provide indirect detection through fluid dynamic signatures, the present study focuses on direct visual inspection supported by deep learning, which can be regarded as complementary to transient analyses in integrated and multimodal monitoring frameworks.

2.1. Signal-Based Approaches

A considerable number of studies focus on analyzing signals such as Acoustic Emission (AE), pressure, or vibration to detect structural anomalies. These methods often involve transforming raw sensor signals into spectrograms or time–frequency representations before classification with CNNs, Long Short-term Memory (LSTM), or hybrid models [2,3,13–15]. While effective in controlled conditions, such techniques require complex sensor networks, often lack spatial localization, and still require improvements in performance in noisy subsea environments [16–18]. Additionally, most are based on simulated data or small proprietary signal sets, which hinders reproducibility [9,19].

2.2. Synthetic and Simulated Visual Data

To overcome the limitations of physical sensors, several authors have proposed the use of synthetic visual data generated from simulations or converted signal spectrograms [2,14,17,20–22]. While this enables the use of established object detection architectures such as You Only Look Once (YOLOv5), DeepLabV3+, or EfficientNet, these studies frequently consider idealized datasets that lack the complexity of real underwater environments, including turbidity, noise, blur, and lighting variation [19,20].

Zhang et al. [23] applied YOLOv5 to synthetic video sequences for leak detection, reporting relevant Mean Average Precision (mAP) scores under controlled conditions. However, the method was not validated on real underwater footage. In a different approach, Zhang et al. [9] transformed acoustic emission signals into image representations to apply 2D CNNs. While effective within that domain, this approach is dependent on calibrated sensors and does not directly leverage the raw visual characteristics of underwater leak events.

2.3. Image-Based Object Detection in Underwater Environments

A growing number of research explores deep learning for object detection using actual underwater imagery, especially for marine life monitoring, coral identification, and general structure tracking [24–26]. Er et al. [27], for example, applied CNNs to underwater object detection with good accuracy on fish and marine debris datasets. However, these applications often work in relatively clear water conditions and are not tailored to the specific visual signatures of pipeline damage or leakage.

Li et al. [16] proposed a hybrid framework combining YOLOv5 and EfficientNet for classifying events in submarine pipeline imagery acquired via ROVs. Although they achieved high detection rates for events such as anodes and debris, their dataset did not include physically replicated leak events, and the generalization capacity of their model to pipeline rupture patterns remains uncertain.

Other relevant efforts, such as Xi et al. [19], include bibliometric reviews of deep learning methods in oil spill detection, which highlight the scarcity of annotated operational image datasets involving pipeline leaks. Most available work still focuses on oil slicks from satellite images or large-scale events, rather than localized, subaquatic failure points.

Despite substantial progress in the field, several challenges remain unaddressed. First, there is a notable scarcity of real, annotated image datasets that represent underwater leak events with sufficient visual detail and diversity. Many existing studies rely heavily on synthetic data or signal-derived representations, which, although useful, do not fully capture the complexity and unpredictability of real subsea environments. Additionally, the robustness of these models to visual artifacts, such as turbidity, lighting variation, and blur commonly encountered in deep-sea conditions, remains limited. Another gap in the current literature is the lack of validated frameworks based on physically simulated leak scenarios, which restricts reproducibility and limits the practical application of these models in the field.

To address these limitations, the present study introduces a dataset generated through controlled experiments in a pressure-regulated tank, where different leak conditions were physically simulated to replicate operational scenarios. This dataset, composed of real underwater imagery, was used to fine-tune and evaluate multiple well-established CNN backbones, including VGG16, ResNet50, InceptionV3, DenseNet121, InceptionResNetV2, and EfficientNetB0, as well as a lightweight custom CNN. The models were tested under visually degraded conditions to ensure applicability in real offshore inspection settings. Notably, the adapted models demonstrated the feasibility of multi-class classification of underwater leaks using only RGB imagery. While performance levels vary across backbones, the approach shows competitive results in validation and provides a realistic benchmark for assessing generalization on independent blind test data.

Thus, this contribution bridges prior conceptual proposals and applied inspection methods by introducing a reproducible framework that integrates experimentally acquired leaks, synthetic imagery for enhanced training, and a blind test based on public NOAA data. This combination enables reproducible benchmarking and more realistic evaluation of deep learning approaches for subsea leak detection.

2.4. Limitations of Conventional Subsea Monitoring and AUV/ROV-Based Inspection

In practical offshore operations, pipeline inspection is frequently performed by AUVs or ROVs equipped with cameras, sonar, or laser scanners. However, most of these systems require manual visual inspection of footage or basic rule-based anomaly detection [23].

Several works have proposed using ROVs for image acquisition but with limited onboard processing or detection capabilities. Traditional visual inspection methods require human operators to review hours of video collected during subsea surveys, a process that is time-consuming, error-prone, and difficult to scale [17].

In addition, even though sonar-based methods or Light Detection and Ranging (LiDAR) systems integrated into AUVs are useful for structure mapping, they remain insufficient for detecting visual cues of localized faults like small leaks, joint corrosion, or early material degradation [24,25]. These systems usually lack the resolution and contextual interpretation required for fine-grained fault classification, especially in environments with high turbidity and poor lighting.

Recent works, such as Schoyen et al. [26], have proposed employing deep learning to automate detection from multibeam echo sounder data, but these approaches still operate at a coarse scale and may fail to integrate contextual image understanding.

The approach proposed in this study complements existing methods by embedding intelligence directly into the visual data stream. This integration opens the possibility of enabling near-real-time leak detection using standard underwater camera systems, including those mounted on ROVs or fixed platforms. Unlike techniques that depend on costly sonar or hyperspectral imaging, the present method relies solely on conventional RGB cameras, reducing equipment complexity and operational costs. Moreover, by automating the detection process end-to-end, this approach has the potential to reduce the human workload and enhance the consistency and reliability of subsea inspections.

By leveraging conventional RGB cameras and automating detection across three classes, the present study addresses these gaps and moves toward practical integration of vision-based leak detection in offshore inspection workflows.

3. Materials and Methods

3.1. Framework Overview

A supervised deep learning framework for detecting underwater pipeline leaks using visual data was developed. The approach is grounded in a proprietary dataset acquired through controlled experiments in a pressurized tank, capturing leak imagery under controlled conditions. These images, together with a set of publicly available media, form the basis for training and evaluating CNNs, with the goal of enabling robust leak detection in noisy, low-visibility subsea environments.

The framework includes four main stages: experimental data acquisition and dataset curation, image preprocessing and annotation, model training and adaptation through fine-tuning of pre-trained CNNs, blind tests, and evaluation using standard performance metrics.

A schematic overview of the proposed framework is presented in Figure 1, illustrating the data acquisition, preprocessing, model fine-tuning, and evaluation steps.

To systematically assess the contribution of data augmentation and synthetic samples, four training configurations were designed:

1. Baseline with experimental data only (no synthetic data, no augmentation),
2. Experimental data plus synthetic images (no augmentation),
3. Experimental data with augmentation only (no synthetic data),

4. Experimental data complemented with both synthetic images and augmentation.

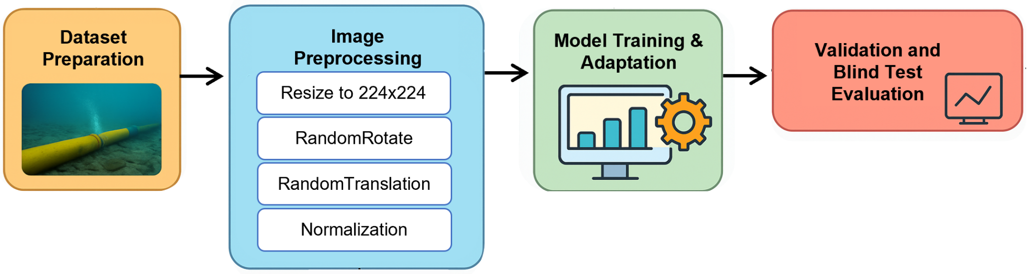


Figure 1. Overview of the proposed deep learning framework for underwater pipeline leak detection.

This design makes it possible to isolate the effect of each strategy, allowing direct comparison of their relevance to model generalization. By evaluating the same CNN backbones under all four setups, we provide insight into whether performance improvements arise primarily from synthetic data, from augmentation, or from their combination, rather than attributing gains to a single factor. Such systematic comparison is particularly important in underwater applications, where collecting large experimental datasets is expensive and time-consuming, and researchers must often decide whether to rely on augmentation, synthetic generation, experimental data acquisition, or combinations of these strategies.

3.2. Experimental Data Acquisition

Leak scenarios were physically simulated in a mesoscale water tank fitted with subsea pipeline sections. Variables such as defect size (e.g., micro-cracks, holes), fluid pressure, and lighting conditions were adjusted to simulate a realistic range of visual leak patterns (e.g., bubbling, jets, discoloration). An underwater RGB camera system was used to capture both still images and video, resulting in a dataset of valuable annotated images.

An underwater RGB action camera (Atrio 4K Wi-Fi, 16 MP, 1920 × 1080 px at 30 fps) was used to capture both still images and video, resulting in a dataset of valuable annotated images. This camera was initially positioned facing the leak simulation pipeline, providing a frontal view of the leak, then repositioned to an orientation perpendicular to the pipeline, capturing the lateral view. These two image acquisition positions are illustrated in Figure 2.

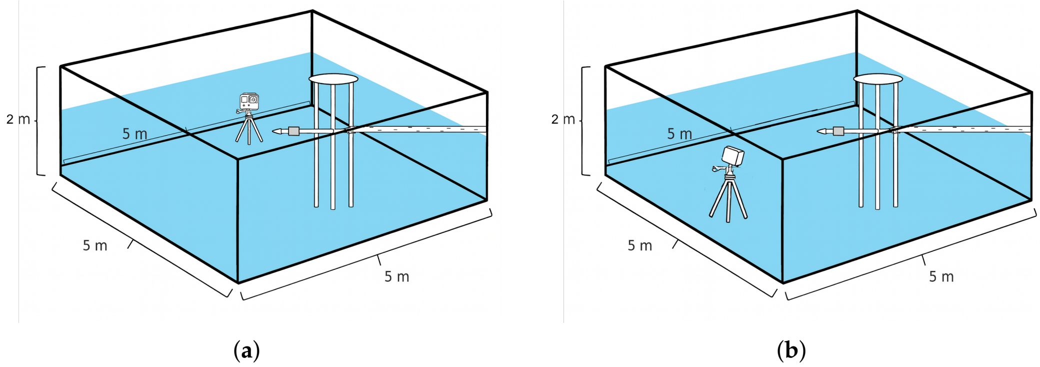


Figure 2. Setup of the experiment with Camera Atrio 4K Wi-Fi, RGB 16 MP, 1920 × 1080 px at 30 fps, in a test tank with internal dimensions of 5 m length × 5 m width × 2 m depth. (**a**) Camera at Position A, facing the leak simulation pipeline, providing a frontal view of the leak. (**b**) Camera at Position B, oriented perpendicular to the pipeline, capturing the lateral view.

A total of 699 images were acquired under controlled leak and no-leak conditions in a pressure-regulated tank. Within this set, 350 correspond to no-leak conditions and 349 to leak events (186 liquid and 163 gas). For training and evaluation, balanced subsets were derived from these images, resulting in different dataset sizes depending on the experimental configuration (see Results and Discussion section). For example, Configurations 1 and 3 relied solely on experimental data, comprising 252 images in total (210 for training and 42 for validation), while Configurations 2 and 4 combined experimental and synthetic images, resulting in 888 images in total (708 for training and 180 for validation).

All images were manually labeled based on visual inspection and experimental condition logs during data acquisition. The complete dataset has been made publicly available through Zenodo [28].

Figures 2 and 3 show the experimental data acquisition performed at the Subsea Technology Laboratory—Federal University of Rio de Janeiro. Tests with a mixture of water and chemical tracer were performed to simulate a liquid leakage through calibrated orifices with nominal diameters of 12.7 mm (1/2 inch) and 6.35 mm (1/4 inch), respectively. For the gas leakage, oxygen was circulated in a pipeline circuit made of 4-inch Schedule 40 carbon steel pipes, with a total length of approximately 100 m, producing bubbles in the water. The test tank used for image acquisition had internal dimensions of 5 m length × 5 m width × 2 m depth.

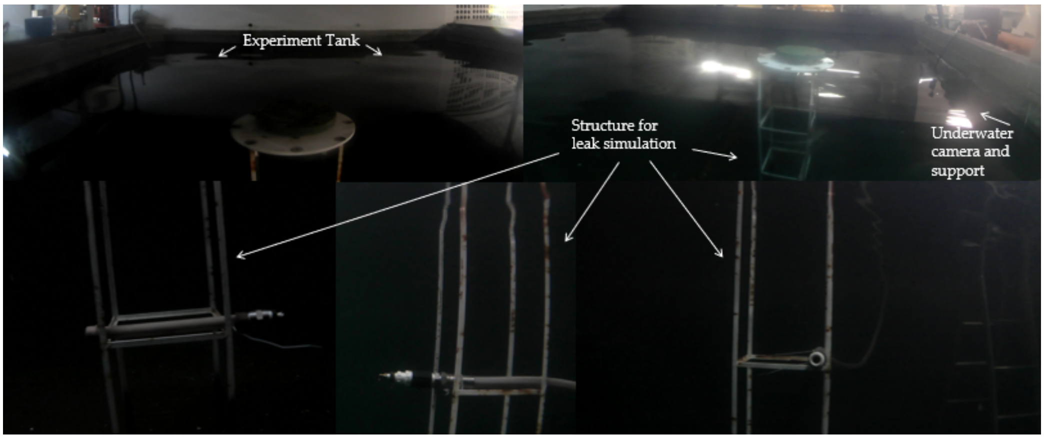


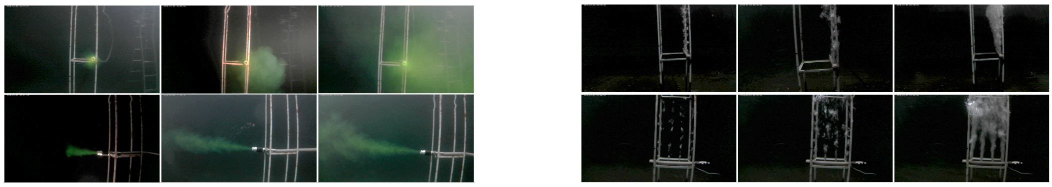
Figure 3. Experimental setup in the mesoscale water tank with pipeline and metal structure to simulate underwater leaks.

To simulate liquid leakages, a small quantity of fluorescein tracer (approximately 100 g) was used to enhance visual contrast in water. Fluorescein is a widely used dye in scientific and industrial applications, including medical diagnostics and flow tracing, due to its high solubility, low toxicity, and detectability at low concentrations [29]. A low-concentration solution was injected through calibrated orifices to form visible plumes, while minimizing residue or staining of tank surfaces. The tracer was pre-tested to ensure detectability under underwater lighting conditions and to avoid excessive coloration of the tank volume.

This experimental dataset formed the baseline training corpus in all configurations, with additional synthetic samples and augmentation applied depending on the training setup.

These images, acquired under controlled underwater conditions, represent a novel resource for visual fault detection research, as no publicly available imagery of experimental underwater leaks has been identified in the literature to date. The data are intended to support reproducible benchmarking in future studies, addressing a critical need in computer vision research where public datasets are essential for consistent model evaluation and comparison.

Figure 3 illustrates the experimental setup and camera positioning, while Figure 4 presents representative images acquired during liquid and gas leak tests.



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

Figure 4. Representative experimental imagery acquired for dataset construction: (**a**) illustrates liquid leak scenarios with fluorescein dye under different lighting and turbidity levels, producing plumes of variable density and visibility; (**b**) shows gas leak scenarios characterized by bubble plumes, irregular structures, and low-contrast conditions. Together, these cases highlight the diversity of visual signatures encountered in underwater leak events and the challenges they pose for CNN training.

3.3. Synthetic Data Generation

The scarcity of publicly available annotated underwater leak images imposes significant challenges for deep learning models, particularly in capturing rare or extreme conditions such as low visibility, strong turbidity, or gas leaks. To address these limitations, we complemented the real datasets with synthetically generated samples designed to increase variability and balance class distribution.

Importantly, synthetic images were used exclusively in training Configurations 2 (experimental + synthetic) and 4 (experimental + synthetic + augmentation), never in blind tests, to ensure unbiased evaluation.

The use of synthetic data has been reported in different marine and engineering domains. For example, Baressi Šegota et al. [30] demonstrated the feasibility of synthetic tabular data for marine turbine exergy analysis, while Kang et al. [31] applied oversampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and autoencoders to seawater temperature profiles, improving anomaly detection in imbalanced datasets. In the visual domain, Bao et al. [32] employed generative frameworks for controlled image synthesis, highlighting their suitability for modular workflows and reproducible pipelines. Shakhovska [33] further proposed a high-fidelity synthetic image generation framework based on Three-Dimensional (3D) representations, reinforcing the role of synthetic visual data in supporting computer vision training when real samples are limited.

In this work, two complementary strategies were applied. First, classical data augmentation was performed on real leak and non-leak samples, including random rotations, flips, Gaussian blur, brightness variations, and Gaussian noise addition. These transformations aimed to simulate degraded underwater conditions and lighting variability. Second, diffusion-based generative synthesis was performed using Stable Diffusion, a latent diffusion model that combines a Variational Autoencoder (VAE), a denoising U-Net [34], and a Contrastive Language–Image Pretraining (CLIP) text encoder for conditioning [35]. The process is controlled by parameters such as sampler type, number of steps, Classifier-Free Guidance (CFG) scale, seed, and resolution, which jointly determine the trade-off between image fidelity, diversity, and computational cost. Extensions to latent diffusion, such as Low-Rank Adaptation (LoRA) [36] and ControlNet [37], enable lightweight fine-tuning and structural conditioning, respectively, although these were not employed in the present study.

For workflow orchestration, we adopted the Comfy User Interface (ComfyUI) framework [38], an open-source, node-based graphical interface that facilitates modular configuration and reproducibility of diffusion pipelines. ComfyUI does not alter the underlying generative model but provides a transparent environment to configure and document the entire workflow. All generated samples were visually inspected, and only those consistent with realistic subsea leak patterns were included in the training set. For transparency, representative prompts, parameter settings, and resulting synthetic images are provided in the Supplementary Material.

While synthetic data cannot replace real-world observations, prior studies confirm their potential to enrich training datasets and improve robustness under constrained conditions [30–33]. Here, synthetic images served exclusively as a complementary strategy to diversify training scenarios, ensuring that final model performance was assessed solely on unseen real images.

It is important to emphasize that blind test sets contained only real underwater images to ensure unbiased model evaluation.

3.3.1. Latent Diffusion Models and Workflow Configuration

The diffusion-based generation of synthetic images in this study relied on Stable Diffusion, a latent diffusion model that combines a Variational Autoencoder (VAE), a denoising U-Net, and a CLIP text encoder for prompt conditioning [35]. In this framework, the diffusion process progressively removes Gaussian noise from a latent representation, reconstructing images guided by text embeddings. Key parameters controlling the process include the sampler type, number of steps, random seed, Classifier-Free Guidance (CFG) scale, and output resolution, which jointly determine the trade-off between image fidelity, variability, and computational cost.

Extensions to latent diffusion enable further customization when necessary. Low-Rank Adaptation (LoRA) [36] provides efficient fine-tuning by inserting trainable low-rank matrices, allowing lightweight adaptation to specific visual domains. ControlNet [37] introduces structural conditioning by injecting edge maps, depth maps, or segmentation masks into the diffusion process, thereby enabling geometric and contextual control while preserving the pretrained generative capacity.

For workflow orchestration, a ComfyUI framework [38] was employed. This tool does not modify the underlying generative model but provides a transparent environment to combine Stable Diffusion with optional modules such as LoRA and ControlNet. This ensured that all synthetic samples were generated in a controlled and reproducible manner, with complete parameter settings documented and representative examples provided in the next subsection.

3.3.2. Synthetic Data Examples

To enhance transparency and reproducibility, this section provides representative examples of the synthetic images generated using the ComfyUI framework [38]. This example includes the text prompt used for conditioning, the main generation parameters (sampler type, number of diffusion steps, guidance scale, seed, and resolution), and some of the corresponding generated output images.

Synthetic data were generated using the Stable Diffusion checkpoint realisticVisionV60B1_v51HyperVAE.safetensors [39] and the LoRA model deep_ocean_photography_hd.safetensors [40], both publicly available on CivitAI under the CreativeML Open RAIL-M license.

These models were selected because the checkpoint is optimized for photorealistic rendering, ensuring that generated samples preserve natural lighting and texture patterns, while the LoRA was specifically trained for underwater photographic domains, increasing fidelity to subsea scenes. The choice of diffusion-based synthesis is supported by recent advances in the underwater vision literature: Xi et al. [41] provide a comprehensive review highlighting the scarcity of annotated underwater datasets and the growing role of image generation for enhancement and restoration; Wang et al. [42] demonstrate that diffusion models can substantially improve underwater image quality through self-supervised learning; and Lu et al. [43] apply a diffusion framework with knowledge distillation for robust coral detection in complex environments. Together, these studies reinforce the suitability of diffusion-based generative models for producing realistic underwater imagery. In our case, the public availability and permissive licensing of the selected models also ensure transparency and reproducibility, which were key requirements of this study.

These examples are intended to illustrate the variability introduced by the synthetic data pipeline and to demonstrate how prompt wording and parameter settings influence the appearance of leak-like features such as bubble plumes, localized turbidity, and lighting conditions. The full set of synthetic images is not included here due to space constraints, but a subset is made available together with the dataset repository for reproducibility.

Figure 5 illustrates representative synthetic images selected for training. Panel (A) shows a submerged pipeline under clear water conditions with uniform illumination, allowing the model to learn the baseline geometry of the structure. Panel (B) presents a leakage scenario with a visible dark crude oil plume dispersing into the surrounding water, which is crucial for the network to recognize leak signatures under moderate turbidity. Panel (C) depicts gas leakage in the form of a bubble plume emerging from a joint, introducing variability in the appearance of leak events and ensuring the model can generalize across liquid and gaseous discharges. Finally, panel (D) combines adverse underwater conditions, with a dense oil plume rising in a cluttered environment close to the seabed, providing challenging examples of occlusion and low visibility. Together, these cases were selected because they cover distinct visual patterns of subsea leaks and environmental contexts, thereby enriching the training dataset with scientifically relevant variability.

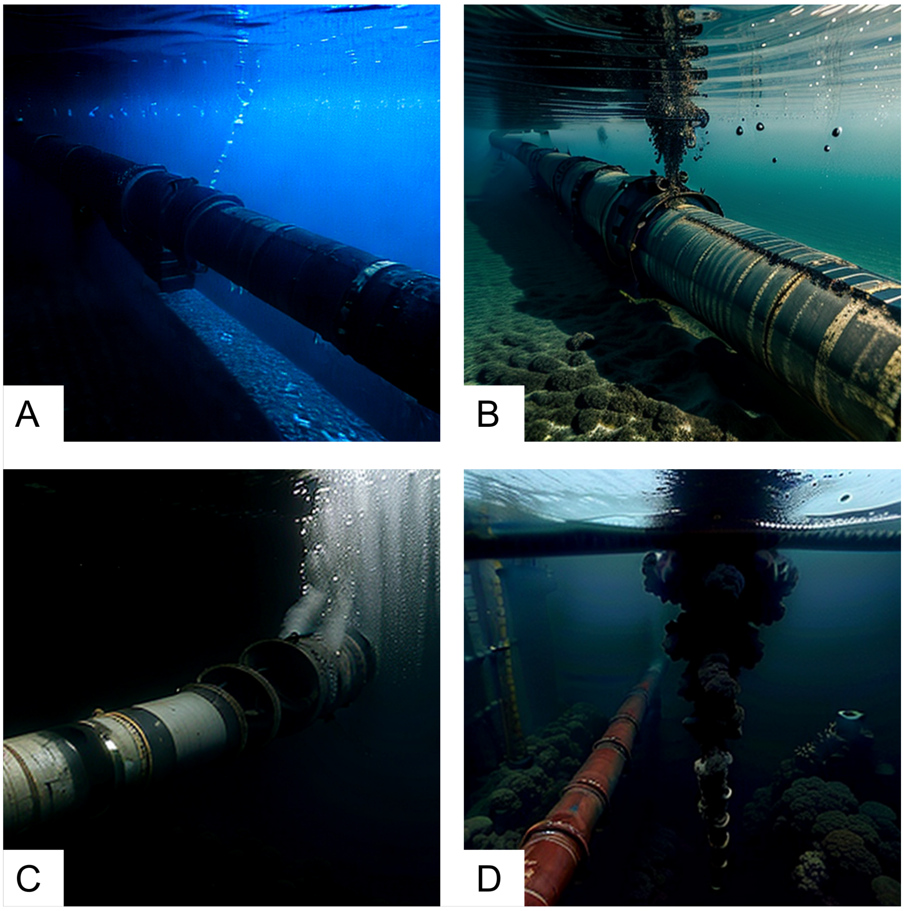


Figure 5. Representative synthetic images generated using Stable Diffusion with parameters described in Table 1 and selected for training: (**A**–**D**) show useful examples that were included in the dataset.

Table 1. Workflow parameters for synthetic image generation (text to image) using Stable Diffusion in the ComfyUI interface.

Field	Value/Setting
Checkpoint and LoRA	
Checkpoint (model)	realisticVisionV60B1_v51HyperVAE.safetensors
LoRA (model)	deep_ocean_photography_hd.safetensors
strength_model	0.50
strength_clip	1.00
Prompts (CLIP Text Encode)	
Positive	underwater oil pipeline leaking black crude oil, visible dark oil plume in water, slow thick leak from pipe joint, realistic underwater scene, seabed; no fire, explosion, surface, or animals.
Negative	text, watermark, blurry, distorted, shiny, surface, sunlight, sky, reflection, people, animals, fish, bright, overexposed, illustration, cartoon, 3D render, glowing, shallow water, cluttered scene, air, gas, lights, fire, explosion.
KSampler	
seed	33559290714832
control_after_generate	randomize
steps	10
cfg	7.0
sampler_name	euler
scheduler	normal
denoise	1.00
Empty Latent Image	
width	512
height	512
batch_size	10
VAE and Output	
VAE Decode	checkpoint default
Save Image: filename_prefix	ComfyUI

Parameters are reported exactly as used in the workflow.

This selection process exemplifies the data curation step that follows model-driven generation, where researchers critically assess the outputs and decide which samples are suitable for training convolutional neural networks and which should be discarded.

Figure 6 shows representative synthetic images that were discarded during dataset curation. Panel (A) was excluded because the gas jet emerging from the seabed exhibits unrealistic geometry and texture artifacts, which could bias the network towards non-physical leak patterns. Panel (B) presents a pipeline segment with insufficient illumination, where the leak region is barely visible, reducing its usefulness for training and potentially introducing label noise. Panel (C), despite containing a pipe structure and bubbles, was discarded due to the exaggerated pipeline geometry and the presence of non-representative background elements, which diverge from real subsea conditions. Finally, panel (D) shows an unrealistic vertical jet directly aligned with the water surface, lacking pipeline context and physical plausibility, thus not contributing to the intended training scenarios. These exclusions were necessary to ensure that the training dataset contained only images with clear physical meaning and representative variability of real subsea leak conditions.

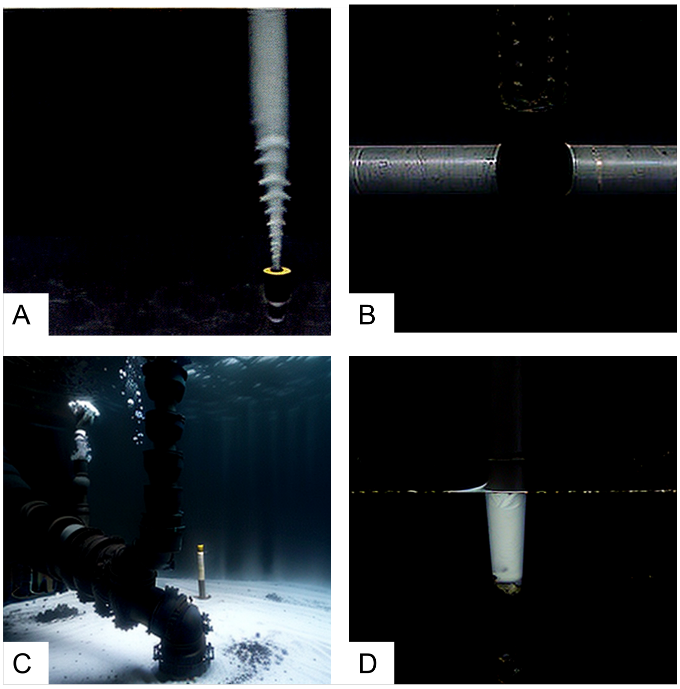


Figure 6. Discarded synthetic images generated using Stable Diffusion with parameters described in Table 1 during dataset curation: (**A**–**D**) show typical cases that were not included due to low realism, artifacts, or inconsistencies.

3.4. Image Preprocessing

All images used in this study were resized to 224 × 224 pixels and normalized. In Configurations 1 (experimental only) and 2 (experimental + synthetic), only resizing and normalization were applied. In Configurations 3 (experimental + augmentation) and 4 (experimental + synthetic + augmentation), the following steps were implemented using the Keras ImageDataGenerator framework [44,45]:

- Horizontal flip: Randomly applied with probability ≈ 50%;
- Rotation: Random rotations up to ±5°;
- Translation: Random shifts up to 8% of the image width and height;
- Zoom: Random zoom in the range of 90–110%;
- Contrast: Random contrast adjustments of ±10%.

These augmentations were applied only to the training set, while validation and blind test sets remained unaltered, ensuring fair model evaluation.

Data augmentation techniques such as rotation and shift are widely used in underwater vision tasks to mitigate the effects of data scarcity and enhance model generalization [46]. These augmentations help models cope with variations in illumination, turbidity, and object positioning challenges commonly encountered in subsea visual inspection.

By explicitly comparing scenarios with and without augmentation, this study provides a clearer assessment of its actual contribution to model robustness, avoiding the common assumption that augmentation is invariably beneficial.

3.5. Model Selection and Training

Multiple well-established CNN architectures were evaluated through transfer learning, including VGG16 [47], ResNet50 [48], InceptionV3 [49], DenseNet121 [50], Inception-ResNetV2 [51], and EfficientNetB0 [52], as well as a lightweight custom CNN designed for computational efficiency. All backbones were initialized with ImageNet-pretrained weights [53], a large-scale image database containing over 14 million labeled images across

1000 object categories, and fine-tuned with the proposed dataset to adapt their learned representations to the specific characteristics of underwater leak detection. This diverse selection of architectures presents benchmark classical baselines, deep residual and densely connected models, scalable architectures, and a domain-tailored lightweight network, providing a comprehensive view of trade-offs between accuracy and computational cost.

The choice of these models was motivated by their complementary characteristics. VGG16 was included as a widely recognized baseline in transfer learning studies, providing a reference point for performance comparison with more advanced architectures. InceptionResNetV2, which combines inception modules with residual connections, was selected to represent an architecture capable of extracting highly discriminative features under challenging underwater conditions such as turbidity and variable lighting. In addition, a lightweight custom CNN was designed specifically for this study to assess the feasibility of a domain-tailored architecture optimized for computational efficiency, an important factor for real-time deployment in AUVs or ROVs with limited onboard processing power. Together, these three models allow a balanced evaluation of baseline, advanced, and task-specific approaches for subsea leak detection [54].

To broaden the benchmarking, additional state-of-the-art CNN backbones were also considered. ResNet50 [48] was included due to its residual learning mechanism, which alleviates vanishing gradient problems and has become a standard reference in computer vision benchmarks. InceptionV3 [49] was chosen for its multi-scale inception modules, which capture visual patterns at different receptive fields, a useful property for detecting heterogeneous leak signatures. DenseNet121 [50] was incorporated as a deeper yet parameter-efficient model, whose dense connectivity promotes feature reuse and mitigates overfitting in limited-data scenarios such as underwater imagery. Finally, EfficientNetB0 [52] was selected for its compound scaling strategy, which balances network depth, width, and resolution to achieve strong accuracy–efficiency trade-offs. The inclusion of these architectures enables systematic comparison across classical baselines, residual and densely connected networks, multi-scale feature extractors, scalable models, and a lightweight custom CNN, thereby providing a comprehensive evaluation of design choices for subsea leak detection.

The architecture can be formally described as:

$$\hat{y} = \mathrm{Softmax}(\mathrm{Dense}(\mathrm{GAP}(f_\theta(x)))) \tag{1}$$

where $f_\theta(x)$ is the CNN feature extractor, Global Average Pooling (GAP), Dense is the fully connected classification head, and Softmax converts logits into class probabilities.

Training used the categorical cross-entropy loss:

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{2}$$

where y_i is the one-hot ground truth and $\hat{y}_i$ is the predicted probability [47].

Models were optimized using the Adam optimizer [55] with early stopping. The blind test set was intentionally kept small to reserve the majority of the data for model training and validation, given the limited dataset size. This subset was used to evaluate model generalization on unseen samples, providing an additional validation point beyond the main training and validation phases.

The CNN models were initially pre-trained on the ImageNet dataset for general image classification. In this study, these architectures were not used as-is; instead, they were fine-tuned and adapted using the custom underwater leak dataset to refine the learned representations for the specific visual patterns associated with subsea faults. This retraining process involved replacing the final classification layers and updating model weights through supervised learning, ensuring that the models could recognize subtle features in real underwater conditions, beyond the scope of the original ImageNet task.

Transfer learning helped reduce overfitting and accelerated convergence, particularly given the limited size of the experimental dataset. This approach is widely adopted in visual tasks with constrained data availability, as demonstrated in comprehensive surveys such as Tan et al. [56].

3.6. Model Architectures Used

VGG16: Introduced by Simonyan and Zisserman [47], this 16-layer deep CNN includes 13 convolutional and 3 fully connected layers. It uses 3 × 3 convolutional filters and 2 × 2 max pooling. Input is an RGB image of size 224 × 224 × 3. The model has about 138 million parameters and is effective for transfer learning on small, domain-specific datasets.

InceptionResNetV2: Proposed by Szegedy et al. [51], this model combines Inception modules [57] with residual connections [48]. It applies parallel convolutional filters of varying sizes and uses pointwise convolutions for efficiency. With about 55 million parameters, it balances depth, performance, and computational efficiency.

ResNet50: Introduced by He et al. [48], this architecture consists of 50 layers and employs residual connections to mitigate the vanishing gradient problem in deep networks. It uses bottleneck blocks composed of 1 × 1 and 3 × 3 convolutions, significantly improving training stability. With approximately 25 million parameters, ResNet50 is widely adopted as a standard backbone in computer vision tasks.

InceptionV3: Proposed by Szegedy et al. [49], this 48-layer network extends the Inception family by factorizing convolutions and incorporating auxiliary classifiers to improve training. Its parallel convolutional filters of different kernel sizes capture multi-scale features efficiently. InceptionV3 has about 23 million parameters and remains a strong reference for image classification benchmarks.

DenseNet121: Introduced by Huang et al. [50], DenseNet121 features 121 layers with dense connectivity, where each layer receives input from all preceding layers. This design encourages feature reuse, reduces the number of parameters, and alleviates vanishing gradients. DenseNet121 has approximately 8 million parameters, making it highly parameter-efficient while achieving competitive accuracy.

EfficientNetB0: Proposed by Tan and Le [52], EfficientNetB0 applies a compound scaling method that uniformly scales network depth, width, and input resolution. Its architecture is based on inverted residual blocks with squeeze-and-excitation optimization. With about 5.3 million parameters, EfficientNetB0 achieves an excellent trade-off between accuracy and computational cost and serves as the baseline of the EfficientNet family.

Custom CNN: A lightweight Sequential model was designed to serve as an efficient baseline for the leak detection under resource-constrained environments. The architecture consists of five convolutional layers with 3 × 3 kernels and ReLU activation, having {16, 512, 512, 216, 216} filters, respectively, each followed by 2 × 2 max pooling. After flattening, three fully connected layers with 64, 64, and 32 units (ReLU) are applied, followed by a final dense layer with 2 units and SoftMax activation. This design was obtained through empirical tuning: starting from a shallow CNN, filter sizes and dense widths were gradually increased while monitoring validation accuracy and generalization. The resulting configuration balanced capacity and performance on the blind test while remaining computationally less demanding than state-of-the-art backbones. This configuration was selected through systematic empirical exploration beginning with lightweight CNNs, then gradually increased depth and filter counts to strike a balance between underfitting and overfitting given dataset size. Although no automated architecture search (e.g., grid, random, or Bayesian search) was conducted, similar approaches based on empirical tuning have been shown to be effective. For example, Deng et al. [58] demonstrate that lightweight custom CNNs designed through empirical adjustments can achieve strong generalization in UAV-based image classification under resource constraints. Likewise, Wojciuk et al. [59] highlight that empirical hyperparameter optimization remains a practical and effective strategy to improve CNN performance when full-fledged automated search is impractical. In our case, the final architecture delivered the best compromise between validation accuracy and generalization on the blind test.

3.7. Training Configuration and Validation Strategy

To train the models efficiently while minimizing overfitting, the Adam optimizer was used with a learning rate of 0.001. This value was selected because it represents a widely adopted default for deep learning tasks, balancing convergence speed and stability due to Adam's adaptive learning rate properties [55]. Preliminary trials with higher learning rates (e.g., 0.01) led to unstable convergence, while lower values (e.g., 0.0001) significantly slowed training. Thus, 0.001 provided the most consistent optimization across all architectures.

Categorical cross-entropy was selected as the loss function because the models were configured with a softmax output layer and three output neurons corresponding to the classes: non-leak, liquid leak, and gas leak. This configuration allowed the use of one-hot encoded labels and categorical loss, ensuring compatibility with pre-trained architectures originally designed for multi-class classification [44].

Each model was trained with a batch size of 32. This choice reflects a trade-off between stability of gradient estimates and computational efficiency: larger batch sizes (e.g., 64 or 128) reduced the frequency of updates and led to worse generalization, while smaller values (e.g., 16) increased training noise without significant accuracy gains. The batch size of 32 is also consistent with common practices in image classification benchmarks, providing a balance between speed and generalization.

Training was conducted for a maximum of 100 epochs to allow sufficient opportunity for convergence while remaining computationally feasible. To avoid overfitting and unnecessary computation, early stopping was applied with a patience of 10 epochs, monitoring validation accuracy and restoring the best weights once performance plateaued. In addition, model checkpointing was used to retain the weights corresponding to the highest validation accuracy achieved during training, ensuring that the best-performing model was preserved. These techniques are established practices for training neural networks effectively while reducing training time and overfitting risk [60].

In the experimental-only scenario, the dataset comprised 210 images for training and 42 for validation, totaling 252 samples balanced across the three classes. In the synthetic-only scenario, the dataset consisted of 498 training images and 138 validation images, totaling 636 samples. When combining experimental and synthetic data, the dataset reached 708 training images and 180 validation images, totaling 888 samples. The blind testing consisted of 30 real images from the NOAA repository. In all cases, the datasets were balanced across the three classes, ensuring equal representation of each failure mode.

The limited dataset size motivated the choice of holdout validation, as applying k-fold cross-validation would reduce the amount of training data available in each fold. Nevertheless, cross-validation remains a valuable technique that could be explored in future work, especially as more data become available [54].

All models were trained on shuffled data, and training was monitored through callbacks to ensure stability. These configurations were selected to balance practical constraints with methodological rigor, enabling consistent evaluation across architectures while respecting the limitations inherent to experimental datasets.

3.8. Evaluation Metrics

To evaluate model performance, standard classification metrics were used: accuracy, precision, recall, and loss. These metrics provide a comprehensive view of how well the models detect leaks versus non-leaks in underwater imagery.

These metrics were computed in the multi-class setting, with per-class precision, recall, and F1-scores reported for non-leak, liquid leak, and gas leak [61]. In addition, macro-averaged values were used as the main comparison metric across backbones and training configurations, providing a balanced view across classes. These aim to provide complementary insights on model behavior in fault detection scenarios [61].

Accuracy was computed as the ratio of correctly classified instances over the total number of predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

where TP denotes true positives (leaks correctly identified), TN true negatives (non-leaks correctly identified), FP false positives (non-leaks incorrectly classified as leaks), and FN false negatives (leaks missed by the model).

Precision quantifies the proportion of true positives among all instances predicted as positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

Recall measures the model's ability to identify all actual leak cases:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

Loss was monitored during training to assess convergence, using categorical cross-entropy as defined previously.

For the three-class setting, precision, recall, and F1-score were computed for each class in a one-vs-rest manner, and the results were aggregated through macro-averaging. F1-score is defined as follows:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Since the dataset was constructed to be balanced (containing an equal number of leak and non-leak images), accuracy was considered an appropriate primary metric. However, precision and recall were also emphasized due to their relevance in practical applications. In leak detection, minimizing false negatives (missed leaks) is crucial, as undetected failures may lead to environmental and operational hazards. Conversely, false positives can trigger unnecessary inspections and delays.

To further analyze classification behavior, confusion matrices were employed, offering insight into error distribution across classes and supporting operational risk assessment. They were derived from model predictions on the blind test set, enabling visual inspection of classification errors and their impact on fault detection reliability. This multi-metric evaluation ensured robust performance analysis beyond accuracy alone, aligned with the critical demands of underwater fault detection systems.

4. Results and Discussion

This section presents the training, validation, and blind testing results for seven CNN architectures: VGG16, ResNet50, InceptionV3, DenseNet121, InceptionResNetV2, EfficientNetB0, and a lightweight custom CNN. Performance metrics and confusion matrices are used to assess each model's ability to classify underwater images into three classes: gas leak, liquid leak, and non-leak.

4.1. Dataset Summary

The datasets used in this study were constructed from experimentally generated leak imagery, with an additional scenario including synthetic data. Two configurations were considered. In the first, the dataset comprised 252 images, balanced across the three classes, with 210 images used for training and 42 for validation. In the second configuration, the same experimentally acquired images were retained and complemented with synthetic samples, resulting in a dataset of 888 images, evenly balanced across the three classes (non-leak, liquid leak, and gas leak), with 708 images used for training and 180 for validation. By subtracting the experimental portion from this combined set, it is possible to isolate the synthetic contribution, which accounts for 636 images, divided into 498 training samples (166 per class) and 138 validation samples (46 per class). Table 2 summarizes the dataset for each case.

Table 2. Number of images per dataset split used in each training configuration. Classes are balanced across gas leak, liquid leak, and non-leak. Config. 2 represents only synthetic images, while Config. 4 corresponds to the combined experimental + synthetic dataset (with augmentation applied during training). Configs. 1 and 3 rely solely on experimental data, with or without augmentation.

Configuration	**Train**	**Validation**	**Test**	**Total**
Config. 1—Experimental Only (No synthetic, No augmentation)	210	42	30	252
Config. 2—Experimental + Synthetic (No augmentation)	498	138	30	636
Config. 3—Experimental + Augmentation (No synthetic)	210	42	30	252
Config. 4—Experimental + Synthetic + Augmentation	708	180	30	888

In both cases, an additional blind test set of 30 images (10 per class) was held out and never used during training or validation, serving exclusively for the final evaluation of generalization performance. This design ensured that differences in performance could be directly attributed to the contribution of synthetic data, while preserving equal class representation and a separate unseen subset for robust assessment.

The synthetic data scenario was designed to extend the experimental dataset rather than replace it. Specifically, the same experimentally acquired images were used in both configurations, with synthetic images added only in the second case. This scenario is part of the four training configurations considered in this study (no synthetic/no augmentation, synthetic only, augmentation only, and synthetic + augmentation), which together provide a controlled basis for assessing the relative contribution of each strategy to model generalization.

It is important to note that data augmentation does not create a new dataset, but rather applies transformations (e.g., rotation, shift, brightness jitter) to the original training images during the learning process. Thus, unlike the synthetic data configurations, the number of training samples remains unchanged; only their variability as seen by the model is increased. This approach is consistent with standard practices in computer vision, where augmentation has long been used to improve generalization under limited data conditions [62,63].

4.2. Performance Comparison of CNN Architectures

To assess the impact of synthetic data and augmentation, four experimental training configurations were evaluated:

- Without synthetic data, without augmentation (Configuration 1).

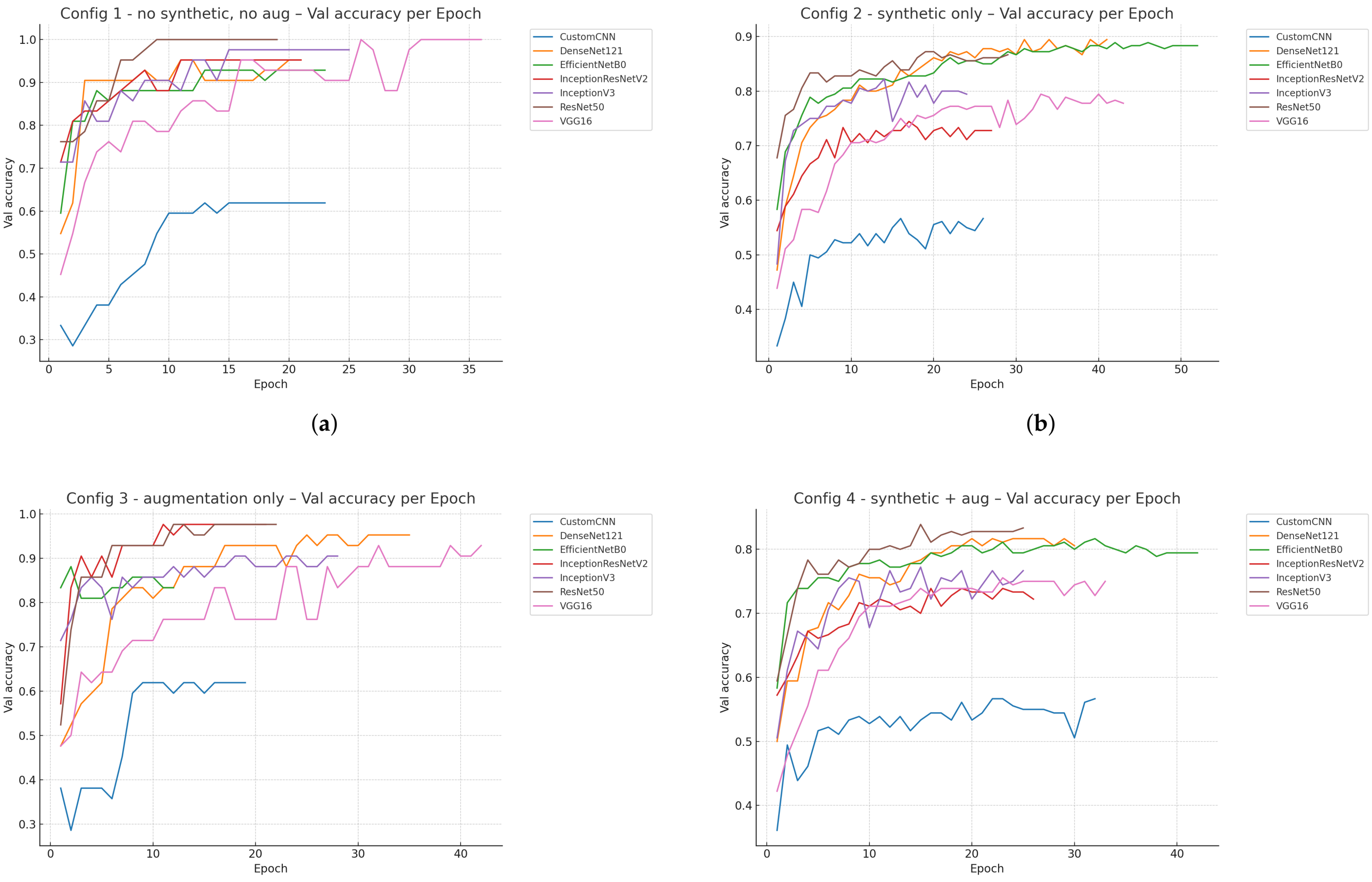
- With synthetic data, without augmentation (Configuration 2).
- Without synthetic data, with augmentation (Configuration 3).
- With synthetic data and augmentation (Configuration 4).

Table 3 represents the validation performance across all training configurations (best checkpoint based on the highest validation accuracy). Figures 7 and 8 show validation accuracy and loss curves across epochs for all backbones, grouped by training configuration.

Appendix A provides full Training and Validation Accuracy and Loss together with the blind test confusion matrices for all backbones and training configurations.

Table 3. Validation performance across all training configurations (best checkpoint based on the highest validation accuracy).

Backbone	Config. 1: No Synthetic, No Aug.			Config. 2: Synthetic Only			Config. 3: Augmentation Only			Config. 4: Synthetic + Aug.		
	Val Loss	Val Acc	Val F1	Val Loss	Val Acc	Val F1	Val Loss	Val Acc	Val F1	Val Loss	Val Acc	Val F1
VGG16	0.259	1.000	1.000	1.054	0.783	0.780	0.337	0.929	0.926	1.147	0.744	0.746
ResNet50	0.207	1.000	1.000	0.468	0.867	0.866	0.243	0.976	0.976	0.487	0.828	0.826
InceptionV3	0.252	0.976	0.976	0.570	0.811	0.810	0.342	0.905	0.903	0.641	0.767	0.763
DenseNet121	0.299	0.952	0.952	0.451	0.894	0.894	0.296	0.952	0.951	0.577	0.806	0.804
InceptionResNetV2	0.264	0.952	0.952	0.653	0.711	0.706	0.281	0.976	0.976	0.666	0.700	0.702
EfficientNetB0	0.312	0.929	0.928	0.453	0.883	0.881	0.429	0.857	0.850	0.543	0.794	0.784
CustomCNN	0.979	0.619	0.532	0.908	0.550	0.541	0.950	0.619	0.532	0.907	0.567	0.554



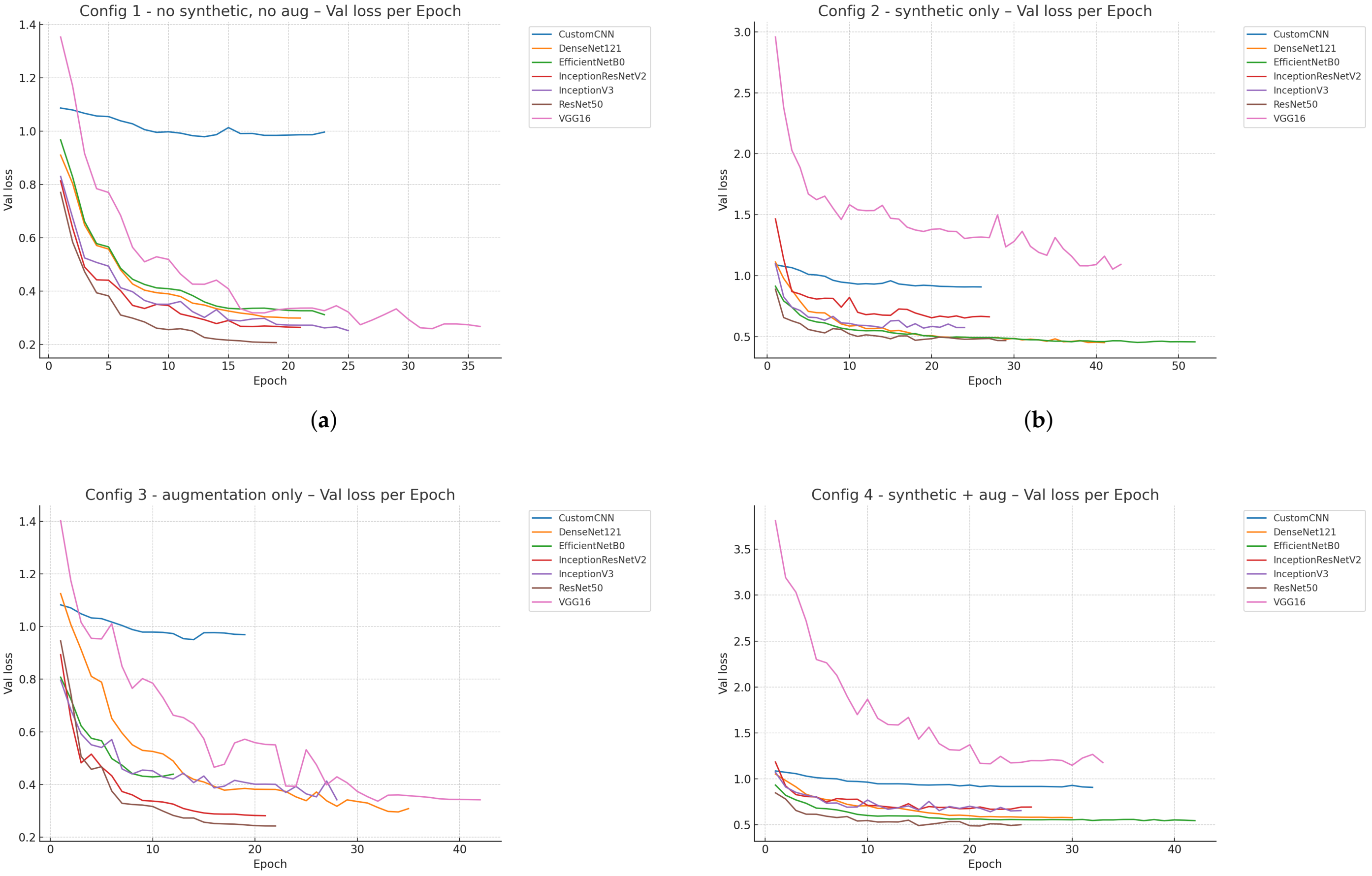
(**a**)

(**b**)

(**c**)

(**d**)

Figure 7. Validation accuracy curves across epochs for all backbones, grouped by training configuration (Config.). (**a**) Config. 1: no synthetic, no augmentation. (**b**) Config. 2: synthetic only. (**c**) Config. 3: augmentation only. (**d**) Config. 4: synthetic + augmentation.



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

(**c**) (**d**)

Figure 8. Validation loss curves across epochs for all backbones, grouped by training configuration. (**a**) Config. 1: No synthetic, No augmentation. (**b**) Config. 2: Synthetic only. (**c**) Config. 3: Augmentation only. (**d**) Config. 4: Synthetic + Augmentation.

The four training configurations enabled controlled evaluation of the relative contribution of synthetic data and augmentation. Configuration 1 (no synthetic, no augmentation) served as the baseline, relying exclusively on experimentally acquired images. In this setting, models were able to distinguish leaks from background cases, but recurrent confusion occurred between gas leak and liquid leak. This reflects the intrinsic visual differences of the phenomena: gas leaks generate small and rapidly dissipating bubble plumes with low contrast, whereas liquid leaks produce larger, more persistent visual disturbances that are easier to detect. Thus, the observed misclassifications align with physical expectations and indicate that the networks were not simply memorizing training data.

Configuration 2 (synthetic only) demonstrated that synthetic data stabilized training and mitigated class imbalance. By enriching under-represented classes, particularly gas leaks, the models reduced their tendency to collapse into a single dominant class. Performance metrics improved in terms of class balance, and confusion matrices revealed more reliable separation between gas plumes and background cases. These findings suggest that synthetic data are effective for extending the coverage of experimental datasets, even if the generated images are not perfect replicas of real-world scenes.

Configuration 3 (augmentation only) yielded counterintuitive results. Instead of improving generalization, augmentation reinforced plume-like structures that were already dominant in the training set. As a result, ambiguous cases were often mapped to the liquid leak class, reducing the sensitivity to gas leaks. This illustrates a key limitation of augmentation when used in isolation: transformations such as rotations, shifts, or brightness adjustments increase visual diversity, but do not create fundamentally new patterns for under-represented classes. Consequently, augmentation amplified existing dataset biases rather than compensating for them.

Configuration 4 (synthetic + augmentation) achieved the most consistent and balanced results across backbones. Here, augmentation increased variability, while synthetic data prevented class collapse, enabling more robust generalization. Confusion matrices showed more even distributions across classes, and borderline predictions (confidence values near 0.5) were represented more faithfully. This synergy highlights the complementary roles of synthetic imagery (class enrichment) and augmentation (variability), confirming the importance of combining both approaches in data-limited underwater monitoring tasks.

It is also worth noting that the modest size of the experimental dataset likely constrained model performance. As widely reported in deep learning literature, model generalization improves significantly with larger and more diverse datasets [64,65]. Thus, increasing the volume of real annotated samples would potentially enhance accuracy and class discrimination even further, complementing the gains already observed with synthetic data.

However, collecting large-scale experimental datasets in subsea environments is often costly, logistically complex, and sometimes unfeasible, which reinforces the relevance of studies such as the present work that explore strategies like smaller experiments, synthetic data generation, and augmentation to mitigate data scarcity.

4.2.1. Configuration 1: No Synthetic, No Augmentation

Table 4 demonstrates that, for Configuration 1 (no synthetic, no augmentation), when models were trained exclusively with the experimentally acquired dataset, validation metrics suggested near-perfect performance for some backbones (e.g., VGG16 and ResNet50 reached validation accuracy and F1-scores of 1.0). However, this apparent success did not translate to the blind test. Systematic misclassifications emerged, particularly between the gas leak and liquid leak classes. This pattern indicates that the networks learned to capture generic features of plume-like structures (e.g., turbidity and bubble dispersion), but failed to distinguish the subtle differences between gaseous and liquid leak events.

Table 4. Config. 1 (no synthetic, no augmentation)—training/validation performance at the best validation checkpoint (highest validation accuracy).

Backbone	Best Ep.	Train Loss	Train Acc	Train Prec	Train Rec	Val Loss	Val Acc	Val Prec	Val Rec	Val F1
VGG16	32	0.373	0.910	0.913	0.905	0.259	1.000	1.000	1.000	1.000
ResNet50	19	0.246	0.981	0.981	0.981	0.207	1.000	1.000	1.000	1.000
InceptionV3	25	0.240	0.986	0.986	0.981	0.252	0.976	0.976	0.952	0.976
DenseNet121	21	0.340	0.919	0.932	0.910	0.299	0.952	0.951	0.929	0.952
InceptionResNetV2	21	0.286	0.957	0.957	0.952	0.264	0.952	0.976	0.952	0.952
EfficientNetB0	23	0.294	0.957	0.971	0.957	0.312	0.929	0.927	0.905	0.928
CustomCNN	13	0.851	0.629	0.663	0.329	0.979	0.619	0.545	0.286	0.532

Confusion with the non-leak class was also observed. In models such as DenseNet121 and EfficientNetB0, background patterns with noise, poor illumination, or seabed textures were misclassified as leaks. The lightweight CustomCNN presented an opposite behavior, showing a strong bias toward the liquid leak class and rarely predicting non-leak, thus over-detecting leaks and ignoring background variability. These behaviors underscore the limited representativeness of training solely on experimental data.

Another limitation of this configuration was the poor calibration of predictive confidence. Several backbones produced highly confident wrong predictions, such as classifying non-leak samples as liquid leak with probabilities above 0.95. Although InceptionV3 and

EfficientNetB0 occasionally provided intermediate probability values (0.4–0.7) for ambiguous cases, most models exhibited overconfidence in their errors, which is problematic in safety-critical subsea applications.

Architectural comparisons reinforce these findings. VGG16 and ResNet50 learned well the internal dataset distribution but generalized poorly. Inception-based models showed more cautious predictions, yet still confused gas with liquid leaks. DenseNet121 and EfficientNetB0 offered more balanced trade-offs, although both struggled with the non-leak class. The CustomCNN was computationally efficient but strongly biased, making it unsuitable for deployment without complementary models.

Overall, this configuration exposed three critical weaknesses: overfitting to the training/validation split, lack of discriminative power between liquid and gas leaks, and frequent false positives in non-leak conditions. These limitations demonstrate that relying solely on limited data is insufficient for robust underwater leak detection. The results highlight the necessity of introducing additional variability through new experiments, synthetic imagery, or augmentation to mitigate overfitting and improve class separation in subsequent configurations.

4.2.2. Configuration 2: Impacts of Adding Synthetic Data

From Table 5, it can be noted that the introduction of synthetic data produced consistent improvements in model robustness, particularly when comparing validation and blind test performance.

Table 5. Config. 2 (synthetic only)—training/validation performance at the best validation checkpoint (highest validation accuracy).

Backbone	Best Ep.	Train Loss	Train Acc	Train Prec	Train Rec	Val Loss	Val Acc	Val Prec	Val Rec	Val F1
VGG16	42	0.512	0.876	0.885	0.869	1.054	0.783	0.782	0.778	0.780
ResNet50	29	0.267	0.975	0.979	0.969	0.468	0.867	0.884	0.844	0.866
InceptionV3	19	0.347	0.929	0.940	0.908	0.570	0.811	0.849	0.783	0.810
DenseNet121	41	0.364	0.921	0.936	0.904	0.451	0.894	0.902	0.867	0.894
InceptionResNetV2	24	0.422	0.891	0.911	0.879	0.653	0.711	0.772	0.694	0.706
EfficientNetB0	45	0.285	0.968	0.969	0.956	0.453	0.883	0.890	0.850	0.881
CustomCNN	24	0.844	0.606	0.681	0.439	0.908	0.550	0.594	0.350	0.541

In the configuration without synthetic data, some backbones (e.g., VGG16, ResNet50) reached nearly perfect validation scores, but failed to generalize to blind test samples, indicating overfitting to the experimental dataset. By contrast, when synthetic images were included, validation metrics decreased slightly, but blind test results became more aligned with observed training performance. This shift demonstrates that synthetic data reduced overfitting and improved the capacity to generalize.

A key distinction emerged in the classification of gas leak versus liquid leak. Without synthetic images, most models exhibited strong confusion between these two classes, frequently predicting gas events as liquid leaks. With synthetic data, class separation improved, especially for DenseNet121 and InceptionResNetV2. Although confusion still occurred, prediction probabilities were more moderate (0.4–0.6), indicating that the networks recognized the ambiguity rather than producing overconfident errors. This is a desirable behavior for operational deployment, as it enables threshold calibration or reject strategies.

Performance on the non-leak class also benefited from synthetic augmentation. In the purely experimental setup, several models—including CustomCNN—systematically misclassified non-leak images as leaks, leading to a high false positive rate. With synthetic data, recognition of the non-leak class improved considerably. Backbones such as VGG16, EfficientNetB0, and ResNet50 assigned more balanced probabilities (0.6–0.8) to background scenes, reducing the tendency to over-detect leaks.

Confidence calibration further confirmed this trend. In the baseline configuration, misclassifications were frequently made with very high probabilities (0.9–1.0), reflecting poorly calibrated models that were “certain” of incorrect decisions. When synthetic data were included, errors persisted but with more moderate confidence values (0.4–0.7). This improvement is practically important: it allows the definition of adjustable thresholds and the implementation of reject options for uncertain predictions.

Architectural comparisons reinforce these findings. DenseNet121, ResNet50, and InceptionResNetV2 benefited most from synthetic data, showing robust and balanced performance across classes. VGG16 maintained high accuracy but with less overfitting, while EfficientNetB0 presented the clearest improvements in confidence calibration, particularly reducing false positives in non-leak cases. The lightweight CustomCNN also improved by no longer ignoring the non-leak class, though it remained less stable than transfer learning backbones.

In summary, the use of synthetic data did not guarantee perfect validation metrics, but it yielded models that were more balanced, better calibrated, and more reliable under blind test conditions. This robustness is more valuable for practical subsea monitoring applications than the artificially inflated performance observed when training only on real experimental data.

4.2.3. Configuration 3: Augmentation Only

Table 6 demonstrates that when augmentation was applied in the absence of synthetic data, the results diverged from expectations. Although validation accuracy and F1-scores remained high for most backbones, blind test results revealed a recurrent collapse of predictions into the liquid leak class. This outcome reflects a reinforcement of dataset biases: geometric and photometric transformations (rotations, shifts, brightness jitter) increased variability within existing classes but did not introduce fundamentally new features, especially for under-represented gas leak cases.

As a consequence, models tended to amplify plume-like cues already dominant in the dataset, leading to oversimplification of the decision boundary.

Table 6. Config. 3 (augmentation only)—training/validation performance at the best validation checkpoint (highest validation accuracy).

Backbone	Best Ep.	Train Loss	Train Acc	Train Prec	Train Rec	Val Loss	Val Acc	Val Prec	Val Rec	Val F1
VGG16	32	0.523	0.824	0.828	0.805	0.337	0.929	0.951	0.929	0.926
ResNet50	22	0.278	0.976	0.976	0.971	0.243	0.976	1.000	0.976	0.976
InceptionV3	28	0.311	0.933	0.946	0.924	0.342	0.905	0.927	0.905	0.903
DenseNet121	34	0.326	0.943	0.947	0.929	0.296	0.952	0.976	0.952	0.951
InceptionResNetV2	21	0.346	0.929	0.928	0.919	0.281	0.976	0.976	0.976	0.976
EfficientNetB0	10	0.437	0.914	0.939	0.881	0.429	0.857	0.919	0.810	0.850
CustomCNN	14	0.834	0.629	0.635	0.448	0.950	0.619	0.562	0.429	0.532

Confusion matrices confirmed that ambiguous samples, including many gas leak instances, were consistently misclassified as liquid leak. In addition, false positives increased for the non-leak class, as background textures and turbidity were exaggerated by augmentation, making them visually closer to leak scenarios. Confidence calibration was also problematic: several misclassifications occurred with high certainty (0.9–1.0), suggesting that augmentation reinforced spurious correlations rather than encouraging caution.

Across backbones, ResNet50 and VGG16 again appeared strong in validation but unstable in blind evaluation. DenseNet121 and EfficientNetB0 showed more resilience, yet still misclassified most gas leaks as liquid. The Custom CNN remained heavily biased toward liquid leak, confirming that shallow architectures are particularly vulnerable to bias reinforcement when augmentation is applied without additional data diversity.

In summary, augmentation alone did not improve generalization. On the contrary, it exacerbated existing imbalances, producing high apparent performance in validation but poor robustness in blind testing. This highlights that augmentation cannot substitute for class enrichment and is most effective when combined with synthetic or real additional samples.

4.2.4. Configuration 4: Synthetic + Augmentation

The combined use of synthetic data and augmentation yielded the most balanced and consistent results. This is demonstrated in Table 7. Validation accuracy and F1-scores were moderate but aligned closely with blind test outcomes, indicating improved generalization. Unlike Configurations 1 and 3, models did not collapse into a single dominant class. Synthetic images enriched under-represented cases, particularly gas leak gas, while augmentation added realistic variability in lighting, orientation, and turbidity. Together, these strategies acted synergistically, reducing bias and promoting robustness across architectures.

Table 7. Config. 4 (synthetic + augmentation)—training/validation performance at the best validation checkpoint (highest validation accuracy).

Backbone	Best Ep.	Train Loss	Train Acc	Train Prec	Train Rec	Val Loss	Val Acc	Val Prec	Val Rec	Val F1
VGG16	30	0.722	0.797	0.815	0.778	1.147	0.744	0.750	0.733	0.746
ResNet50	21	0.354	0.924	0.934	0.912	0.487	0.828	0.852	0.800	0.826
InceptionV3	22	0.448	0.888	0.901	0.859	0.641	0.767	0.804	0.728	0.763
DenseNet121	30	0.469	0.867	0.889	0.838	0.577	0.806	0.829	0.783	0.804
InceptionResNetV2	15	0.595	0.773	0.803	0.739	0.666	0.700	0.718	0.678	0.702
EfficientNetB0	39	0.357	0.925	0.931	0.915	0.543	0.794	0.854	0.778	0.784
CustomCNN	32	0.858	0.607	0.670	0.441	0.907	0.567	0.613	0.378	0.554

Confusion matrices showed that all three classes (gas leak, liquid leak, non-leak) were represented with fewer systematic misclassifications. Errors that remained were typically associated with borderline cases and were accompanied by moderate prediction confidence (0.4–0.7), indicating that the models recognized ambiguity instead of making overconfident mistakes. This calibration is a key advantage for real-world deployment, as it allows the definition of adaptive thresholds or the rejection of uncertain predictions.

Among the backbones, DenseNet121 and ResNet50 achieved the most stable balance between accuracy and calibration. InceptionV3 and InceptionResNetV2 also performed well, though with slightly higher variability in gas–liquid separation. EfficientNetB0 stood out for offering strong performance with reduced computational cost, while VGG16 remained competitive but less efficient. The CustomCNN improved compared to previous scenarios, recognizing all three classes more evenly, though still with lower absolute performance compared to transfer learning models.

Overall, this configuration demonstrated that synthetic data and augmentation are not redundant but complementary. Synthetic images expand class coverage, while augmentation increases variability within each class. The combined effect produced the most reliable generalization among all tested scenarios, supporting the value of a data-centric approach in underwater leak detection tasks.

4.3. Model Architecture Considerations

The evaluation across seven backbones revealed important differences in how architectures handle the underwater leak detection task. Residual and densely connected networks (ResNet50, DenseNet121, InceptionResNetV2) consistently showed stable training behavior, reflecting their ability to propagate features effectively and reuse information in data-limited conditions. Among them, ResNet50 provided a robust and widely recognized baseline, while DenseNet121 offered parameter efficiency and strong validation performance. InceptionResNetV2 achieved competitive accuracy but showed some sensitivity to class imbalance in scenarios without synthetic data.

InceptionV3, with its multi-scale inception modules, captured visual features at different receptive fields and performed well in most cases, although its behavior was less stable for gas-leak detection. VGG16, despite being a canonical reference for transfer learning, presented limitations due to its large parameter count and comparatively rigid structure, which hindered generalization in the multi-class setting. EfficientNetB0 demonstrated a favorable accuracy–efficiency trade-off, achieving solid results with fewer parameters, though its robustness decreased in visually degraded conditions such as turbid or low-contrast imagery. Finally, the lightweight custom CNN confirmed the feasibility of domain-specific models optimized for efficiency, but its reduced depth limited its capacity to capture subtle visual cues, making it less reliable for distinguishing gas leaks from similar background patterns.

Overall, these observations emphasize that while lightweight architectures are appealing for real-time applications in resource-constrained environments (e.g., AUVs or ROVs), deeper networks with residual or dense connectivity currently offer the most reliable performance in underwater leak detection. The systematic benchmarking across diverse backbones provides valuable insight into trade-offs between accuracy, computational cost, and robustness to challenging visual conditions.

4.4. Application to Field Data: Blind Test with Deepwater Horizon Oil Spill and the Atlantic Margin Natural Methane Seeps

To assess the practical applicability of the leak classification models in real-world scenarios, a set of publicly available images from the Deepwater Horizon oil spill in 2010 at the Gulf of Mexico was employed. These field images were sourced from the NOAA Office of Response and Restoration of the United States repository [66] and were not used during training or validation and served as an additional test to assess generalization performance beyond the experimental dataset. A subset of leak images was selected and processed using the trained models and the custom CNN to evaluate classification performance under field conditions for all the four configurations studied.

It is important to note that, due to the lack of publicly available images of verified gas leaks in subsea pipelines, the blind test set for the gas leak class was performed with imagery of natural methane seeps. These images were obtained from NOAA campaigns in the Atlantic margin of the United States, where the ROV Deep Discoverer (D2) explored methane plumes rising from the seafloor [67]. While such seeps are not identical to gas discharges from damaged pipeline sections, they represent the closest available real-world analogue of underwater gas leaks, providing valuable visual references of bubble plumes, turbidity, and dispersion under deep-sea conditions. We acknowledge that this substitution introduces a potential mismatch between training and testing distributions, since the network was trained on controlled pipeline leak experiments rather than natural seepage phenomena. Nevertheless, incorporating these samples was considered scientifically justified, as they expose the models to authentic subsea gas plumes and allow a more realistic assessment of generalization in the absence of dedicated pipeline datasets.

For clarity, a summary of the blind test performance under the best-performing setup, Configuration 4 (synthetic + augmentation), is reported in Table 8. Among the evaluated backbones, InceptionV3 achieved the highest performance, with a macro F1-score of 0.671 and accuracy of 0.667, demonstrating its superior generalization capability in this scenario. Representative prediction outcomes for Configurations 1 to 4 are shown in Figures 9–12.

Table 8. Blind test performance (Configuration 4: synthetic + augmentation). Results are reported as macro-averages for Precision, Recall, and F1-score, along with overall Accuracy.

Model	Precision	Recall	F1-Score	Accuracy
CustomCNN	0.563	0.500	0.459	0.500
DenseNet121	0.349	0.367	0.326	0.367
EfficientNetB0	0.642	0.500	0.467	0.500
InceptionResNetV2	0.384	0.400	0.364	0.400
InceptionV3	0.713	0.667	0.671	0.667

Figure 9 represents blind test predictions for Configuration 1—without synthetic, without augmentation: Note the recurrent confusion between gas leak and liquid leak (images D to F), as well as unstable predictions for background cases (image C). In Configuration 1, models trained solely on the original dataset showed partial ability to discriminate leak types (images A and B), with recurrent confusion between gas and liquid leaks. Although the models captured plume-like structures, their decision boundaries were unstable, leading to inconsistent performance across backbones.

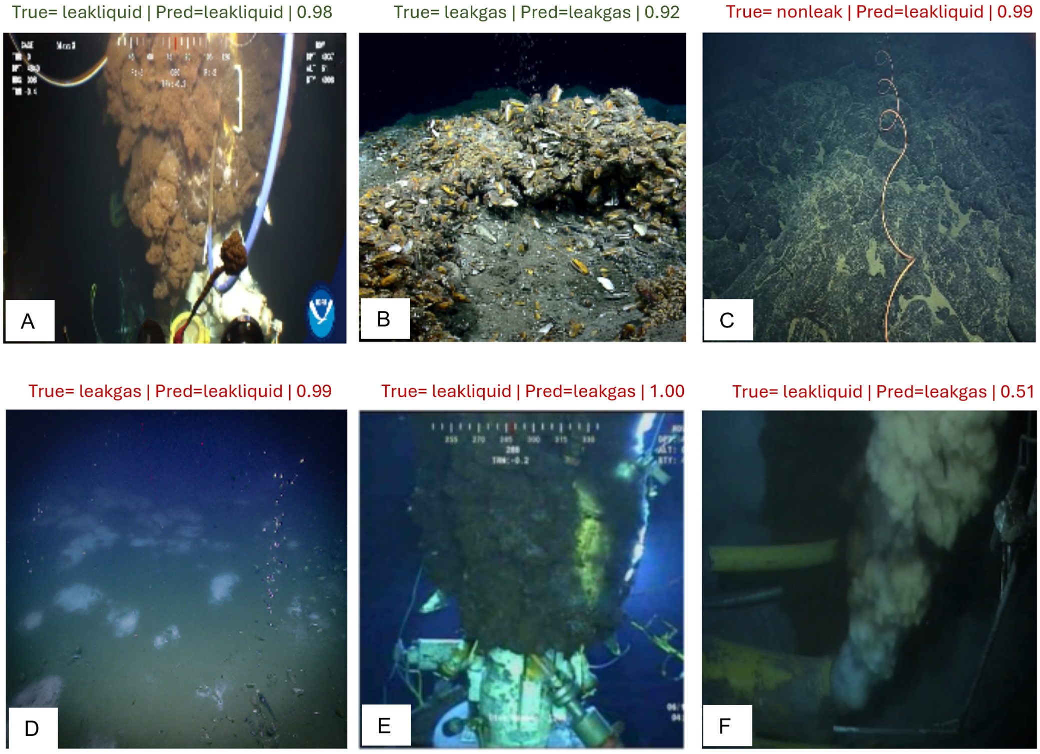

True= leakliquid | Pred=leakliquid | 0.98
True= leakgas | Pred=leakgas | 0.92
True= nonleak | Pred=leakliquid | 0.99
A
B
C
True= leakgas | Pred=leakliquid | 0.99
True= leakliquid | Pred=leakgas | 1.00
True= leakliquid | Pred=leakgas | 0.51
D
E
F

Figure 9. Results of the blind tests for training Configuration 1—only experimental data (no synthetic data and no augmentation). Subfigures (**A**–**F**) illustrate representative cases, including partial discrimination between leak types, confusion between gas and liquid leaks, and unstable predictions for background samples.

The results of blind test predictions for Configuration 2 in Figure 10 demonstrated that synthetic data considerably improved training stability and reduced class imbalance effects. The models trained with synthetic data achieved more robust discrimination of gas plumes (images A, B, D, and F) and background scenes, avoiding collapse into a single class and producing more balanced confusion matrices. There was improved stability and class balance. Synthetic images supported the correct classification of gas leak events that were otherwise misclassified.

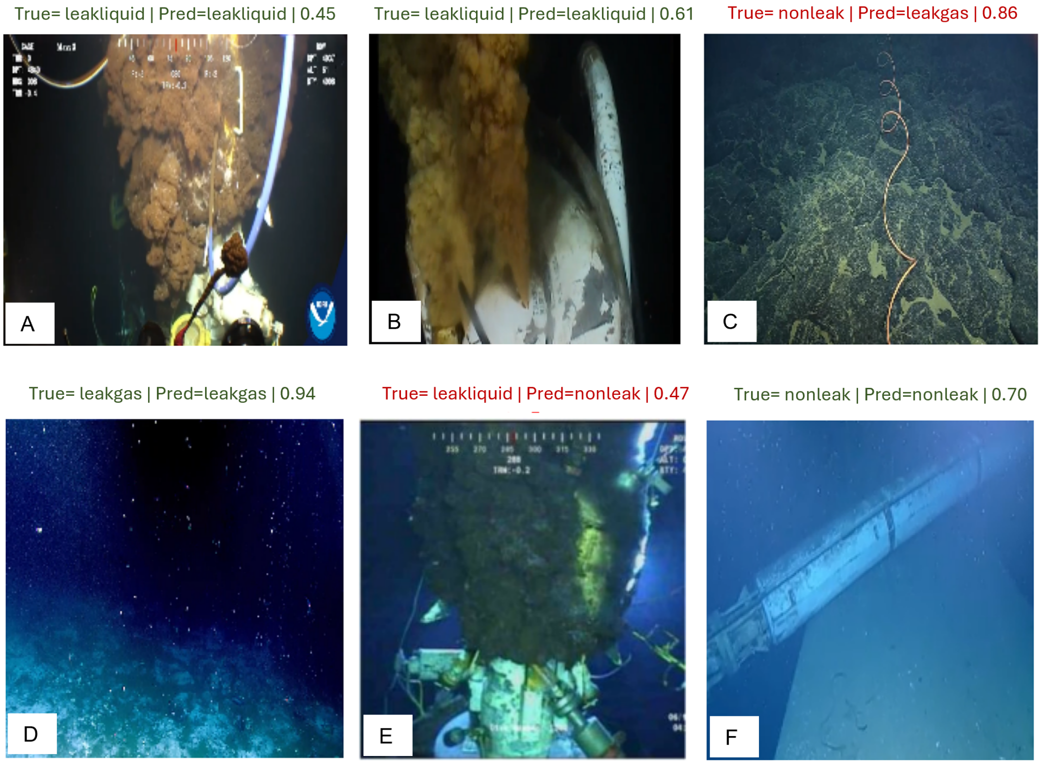

True= leakliquid | Pred=leakliquid | 0.45
True= leakliquid | Pred=leakliquid | 0.61
True= nonleak | Pred=leakgas | 0.86
A
B
C
True= leakgas | Pred=leakgas | 0.94
True= leakliquid | Pred=nonleak | 0.47
True= nonleak | Pred=nonleak | 0.70
D
E
F

Figure 10. Results of the blind tests for training Configuration 2—with synthetic data, without augmentation. Subfigures (**A**–**F**) illustrate representative cases, showing improved discrimination of gas plumes and background scenes, reduced class imbalance effects, and more stable predictions compared to Configuration 1.

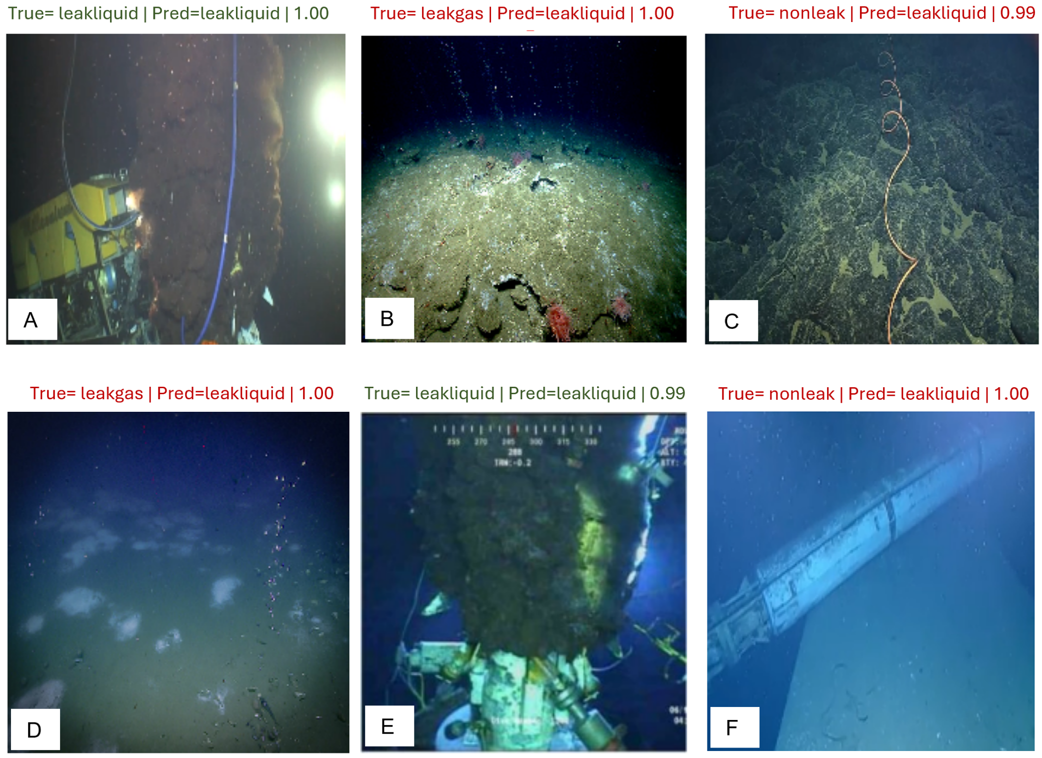

True= leakliquid | Pred=leakliquid | 1.00
True= leakgas | Pred=leakliquid | 1.00
True= nonleak | Pred=leakliquid | 0.99
A
B
C
True= leakgas | Pred=leakliquid | 1.00
True= leakliquid | Pred=leakliquid | 0.99
True= nonleak | Pred= leakliquid | 1.00
D
E
F

Figure 11. Results of the blind tests for training Configuration 3—without synthetic data, with augmentation. Subfigures (**A**–**F**) illustrate oversimplified classification boundaries, with ambiguous samples frequently mapped to the leak–liquid class, reflecting limited generalization from augmentation alone.

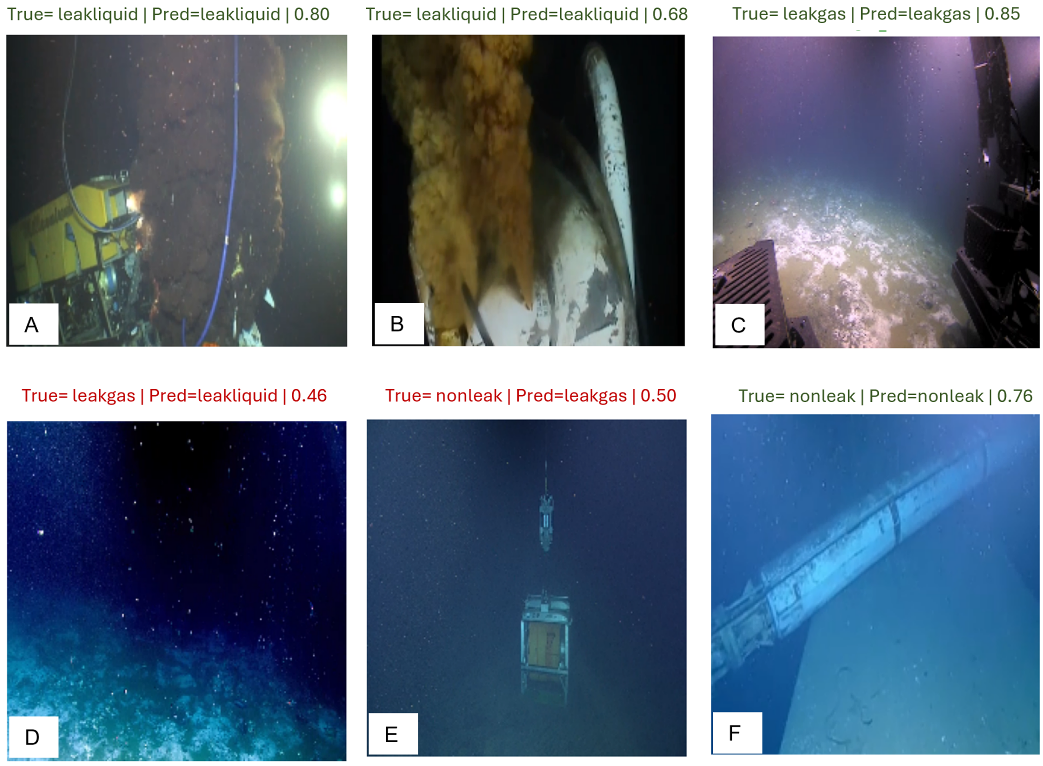

True= leakliquid | Pred=leakliquid | 0.80
True= leakliquid | Pred=leakliquid | 0.68
True= leakgas | Pred=leakgas | 0.85
A
B
C
True= leakgas | Pred=leakliquid | 0.46
True= nonleak | Pred=leakgas | 0.50
True= nonleak | Pred=nonleak | 0.76
D
E
F

Figure 12. Results of the blind tests for training Configuration 4—with synthetic data and augmentation. Subfigures (**A**–**F**) illustrate the most consistent and robust results, with predictions more evenly distributed across classes and borderline cases (**D**,**E**) better represented, highlighting the complementary roles of synthetic imagery and augmentation.

In Configuration 3, whose results can be observed in Figure 11, the introduction of light augmentation operations (rotation, shift, brightness jitter) without synthetic data resulted in a counterintuitive outcome: instead of improving generalization, the models tended to oversimplify the classification boundary (images B, C, D, and F). Many ambiguous samples were consistently mapped to the liquid leak class, indicating that the augmentation reinforced plume-like visual cues without introducing enough diversity for non-leak and gas-leak contexts.

Finally, the results for Configuration 4, demonstrated in Figure 12, yielded the most consistent and robust results. The combination of synthetic data and augmentation enabled the networks to generalize across challenging blind test images. Predictions were more evenly distributed across classes (images A, B, C, and F), and borderline cases (e.g., with confidence 0.5, as in images D and E) were more faithfully represented. This synergy highlights the complementary roles of dataset enrichment through synthetic imagery and augmentation-induced variability.

4.5. Implications and Practical Considerations

The comparative analysis across scenarios provides key insights into the role of data diversity in underwater leak detection tasks.

First, synthetic data proved essential to stabilize training. Models without synthetic samples (Scenarios 1 and 3) often converged to biased solutions, particularly in the presence of augmentation, which exaggerated plume-like structures and caused class collapse into liquid leak. This confirms that augmentation alone cannot address intrinsic dataset imbalance nor compensate for underrepresented leak modes.

Second, augmentation amplified existing biases when applied without synthetic data. The tendency to misclassify ambiguous cases as liquid leak suggests that the operations (blur, intensity jitter, spatial shifts) magnified features already abundant in the dataset, while failing to generate meaningful variations for gas leaks or background patterns. This explains the apparent paradox: augmentation degraded generalization when used in isolation.

Third, the best-performing models emerged from combining synthetic data and augmentation (Configuration 4). In this setting, augmentation acted synergistically with synthetic data by diversifying the visual space while synthetic images prevented class collapse. The result was a more balanced performance across leak types, with representative examples correctly classified even in borderline cases (e.g., Figure 12A,B, where predictions hovered near 0.5).

Overall, these findings indicate that data-centric strategies are indispensable for underwater leak detection. Synthetic data enriches the representation of under-sampled classes, while augmentation, when applied cautiously, further enhances robustness. Future work should investigate adaptive or class-specific augmentation schemes, as well as advanced generative approaches (e.g., diffusion-based synthetic imagery) to further reduce residual misclassifications.

Although the absolute values of validation and blind test accuracy do not reach the highest levels typically reported in large-scale vision benchmarks, the results provide meaningful insights into the challenges of underwater leak detection. In particular, gas leaks proved consistently harder to classify than liquid leaks, reflecting their subtle visual signatures, characterized by small bubble plumes and limited turbidity. This behavior highlights the sensitivity of CNNs to physical properties of the leak phenomena and confirms that the models are learning realistic patterns rather than overfitting spurious cues.

A comparison between validation and blind test performance provides additional insight into the generalization capacity of the models. Across backbones, validation accuracy and macro F1-scores were generally higher than those obtained on blind test samples, which is expected given the limited dataset size and the inherent variability of unseen images. However, the blind test results did not collapse to random guessing, indicating that the networks learned meaningful representations of leak phenomena. For example, models that achieved strong validation stability under Configuration 4 also retained balanced behavior on the blind test set, confirming that the synergy between synthetic data and augmentation improved robustness. The relative consistency of performance trends across validation and blind test evaluations suggests that the models were not overfitting to the training conditions, but rather captured transferable visual cues related to subsea leak events. This observation, while modest in absolute performance, reinforces the feasibility of applying CNN-based approaches to underwater monitoring tasks, even when training data availability is limited.

4.6. Limitations and Future Work

Although deep learning architectures typically require large-scale datasets, the constraints of subsea experimental data collection inevitably lead to smaller sample sizes. To mitigate this challenge, several strategies were adopted. First, transfer learning from ImageNet was employed to leverage generalized visual representations. Second, extensive data augmentation (including geometric and photometric transformations) was applied to artificially increase variability. Third, model robustness was verified using a blind test set composed of previously unseen images, ensuring that performance was not solely a result of overfitting. Additional regularization techniques such as dropout and early stopping further improved generalization [60]. While the dataset size remains a limitation, it reflects the realistic difficulties of acquiring subsea leak imagery under controlled conditions. Importantly, the public release of this experimental dataset aims to support future research, allowing the community to expand training datasets and enhance the generalization capability of vision-based leak detection models.

Overall, these results demonstrate that with appropriate model adaptation and curated data, even classical CNN architectures can effectively detect underwater pipeline leaks, supporting scalable, cost-efficient monitoring strategies in offshore environments.

Future research could also investigate the integration of vision-based deep learning with transient test methodologies. While the former provides direct evidence of leaks through visual inspection data, the latter infers anomalies from hydraulic pressure signatures [11,12]. Leveraging this multimodal approach in a complementary way may enhance the robustness of subsea pipeline monitoring systems and improve their operational reliability in real offshore environments.

5. Conclusions

This study investigated deep learning approaches for subsea pipeline leak detection using experimental and public visual data. The main findings and contributions can be summarized as follows.

First, a novel experimental dataset was generated under controlled underwater conditions, comprising 699 annotated images of leak and no-leak scenarios. This dataset has been made publicly available through Zenodo to support transparency and reproducibility.

Second, six well-established Convolutional Neural Network (CNN) architectures were benchmarked against a custom lightweight CNN specifically designed for computational efficiency in underwater applications.

The results confirmed that transfer learning from large-scale databases such as ImageNet is a viable strategy for subsea leak detection, enabling CNN backbones to adapt to underwater imagery despite the modest size of the experimental dataset. While performance varied across architectures, models such as InceptionV3 and DenseNet121 achieved stable generalization under blind testing, whereas lightweight or heavily parameterized networks showed less consistent behavior. This highlights both the advantages and the trade-offs of applying transfer learning in this domain.

Furthermore, while the experimental dataset formed the basis for training and validation, blind testing on publicly available subsea imagery (NOAA leaks and methane seeps) provided additional diversity and demonstrated model robustness under degraded visual conditions such as turbidity, blur, and low lighting.

Finally, the proposed visual approach provides a cost-effective alternative for early fault detection using standard camera systems, offering potential for integration into real offshore monitoring frameworks.

Future research may focus on multimodal integration with transient hydraulic tests and acoustic sensing to further enhance robustness in complex offshore environments.

Supplementary Materials: All supplementary images and datasets are available in the public repository on Zenodo (https://doi.org/10.5281/zenodo.16899245).

Author Contributions: V.F.d.S. conducted the experiment and the numerical simulation, analyzed the result, and wrote the manuscript. T.A.N. reviewed, modified the manuscript and supervised it. B.A.R. assisted in the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by CNPq (Brazilian National Council for Scientific and Technological Development).

Data Availability Statement: The experimental dataset generated and analyzed during this study has been deposited in Zenodo and is openly accessible at https://doi.org/10.5281/zenodo.16899245.

Acknowledgments: The authors thank COPPE, Federal University of Rio de Janeiro, for their support in conducting this study and the NOAA (National Oceanic and Atmospheric Administration of United States) Office of Response and Restoration for the online availability of Deepwater Horizon Oil Spill and Atlantic Margin Methane Seeps database records.

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

Nomenclature

Latin Letters

F_1	F1-score (harmonic mean of Precision and Recall)
FN	False Negatives (number of samples)
FP	False Positives (number of samples)
L_{CE}	Categorical cross-entropy loss function (dimensionless)
TN	True Negatives (number of samples)
TP	True Positives (number of samples)
y_i	Ground truth label for class i (dimensionless)
$\hat{y}_i$	Predicted probability for class i (dimensionless)

Greek Letters

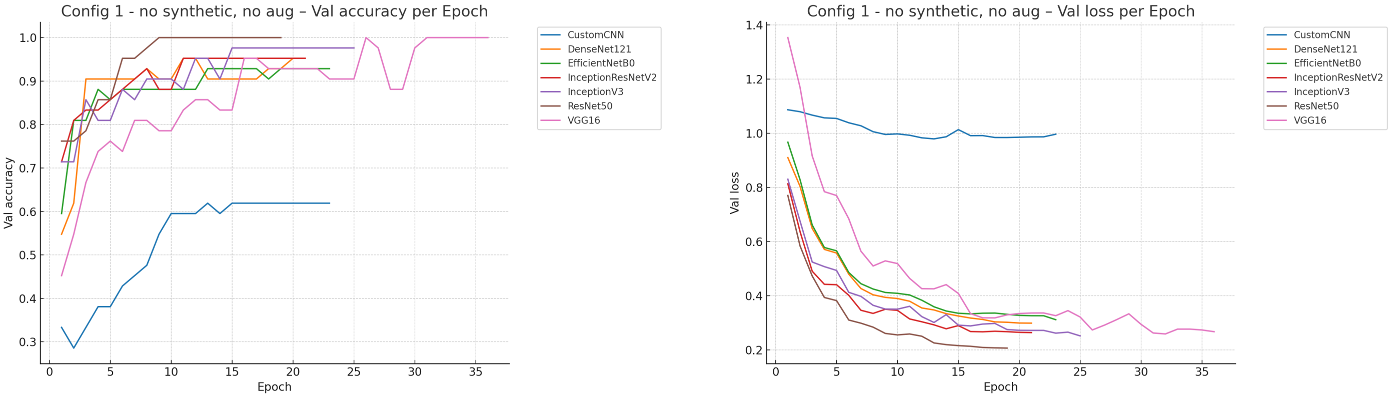
θ	Parameters of the neural network model

List of Acronyms

3D	Three-Dimensional
AE	Acoustic Emission
AUV	Autonomous Underwater Vehicle
CFG	Classifier-Free Guidance
CLIP	Contrastive Language–Image Pretraining
CNN	Convolutional Neural Network
GAP	Global Average Pooling
LiDAR	Light Detection and Ranging
LoRA	Low-Rank Adaptation
mAP	Mean Average Precision
NOAA	National Oceanic and Atmospheric Administration
RGB	Red, Green, Blue (color channels)
ROV	Remotely Operated Vehicle
SMOTE	Synthetic Minority Over-sampling Technique
VAE	Variational Autoencoder
YOLO	You Only Look Once (object detection algorithm)

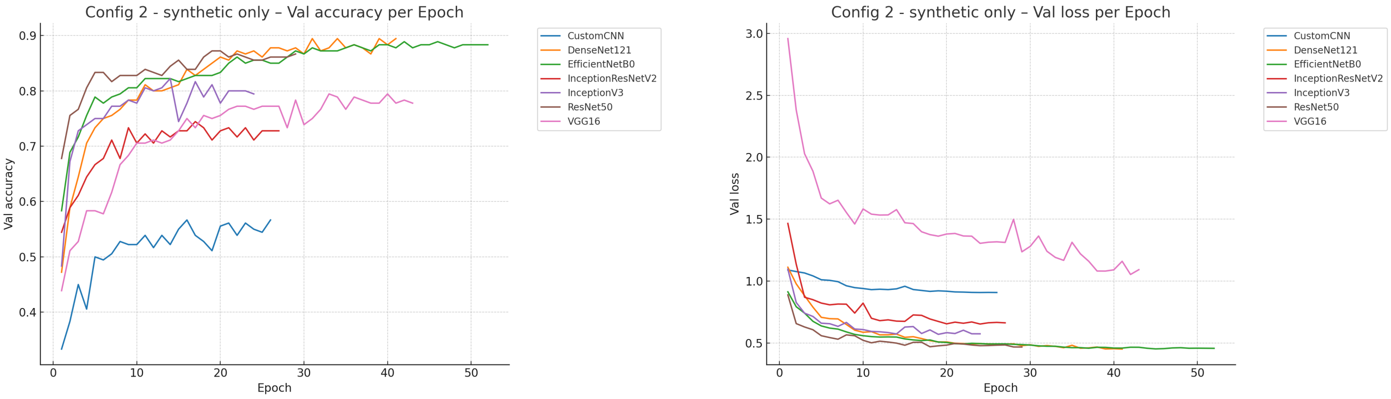
Appendix A. Supplementary Visual Results

This appendix provides supplementary figures illustrating full Training and Validation Accuracy and Loss. The blind test confusion matrices for all backbones and training configurations are also provided.



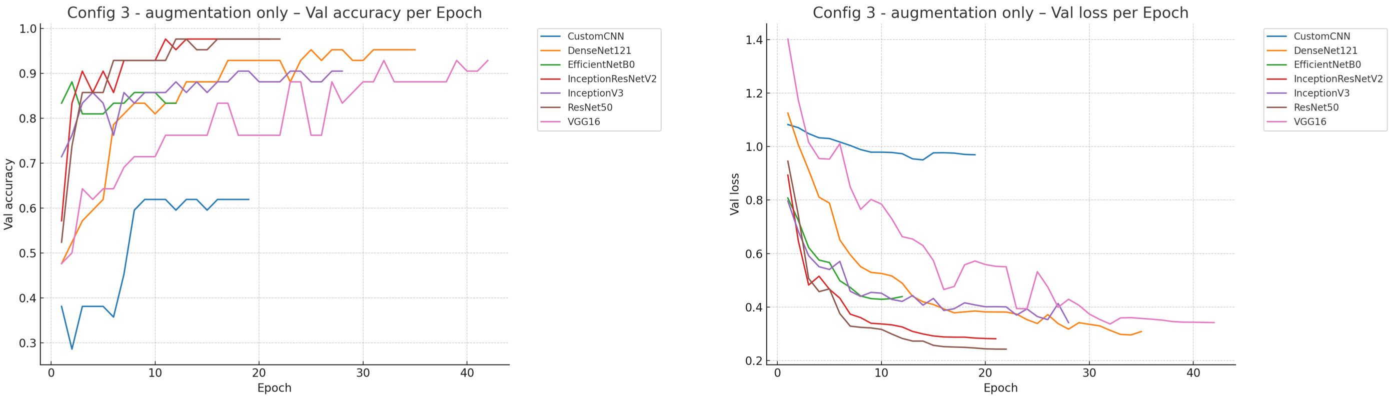
(**a**) (**b**)

Figure A1. Training curves for Configuration 1 (no synthetic, no augmentation): (**a**) validation accuracy and (**b**) validation loss.



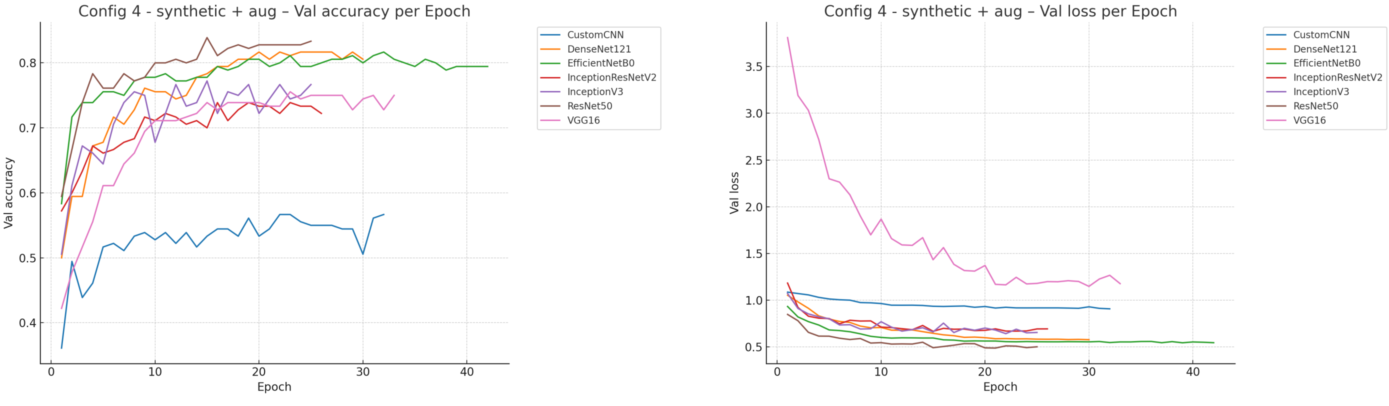
(**a**) (**b**)

Figure A2. Training curves for Configuration 2 (synthetic only): (**a**) validation accuracy and (**b**) validation loss.



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

Figure A3. Training curves for Configuration 3 (augmentation only): (**a**) validation accuracy and (**b**) validation loss.



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

Figure A4. Training curves for Configuration 4 (synthetic + augmentation): (**a**) validation accuracy and (**b**) validation loss.

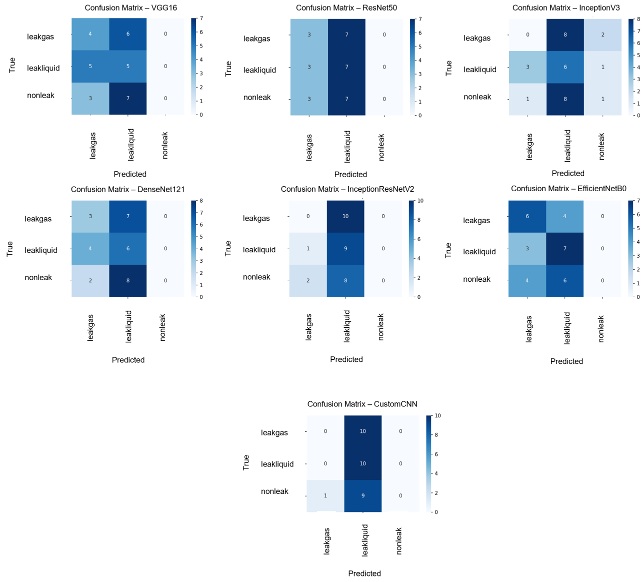


Figure A5. Confusion matrix for Configuration 1 (no synthetic, no augmentation).

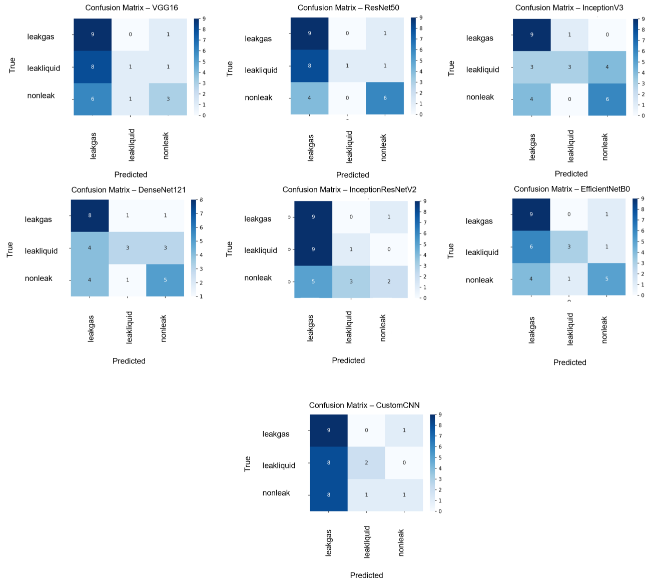


Figure A6. Confusion matrix for Configuration 2 (synthetic only).

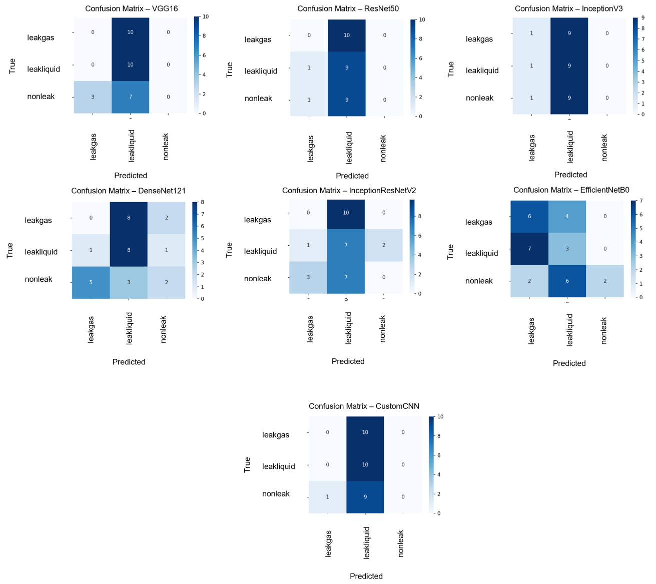


Figure A7. Confusion matrix for Configuration 3 (augmentation only).

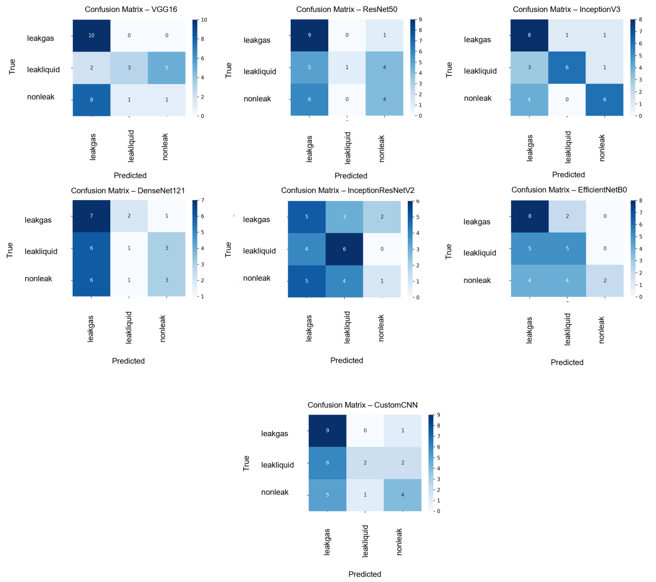


Figure A8. Confusion matrix for Configuration 4 (synthetic + augmentation).

References

1. Dang, T.; Nguyen, T.T.; Liew, A.W.C.; Elyan, E. Event Classification on Subsea Pipeline Inspection Data Using an Ensemble of Deep Learning Classifiers. *Cogn. Comput.* **2025**, *17*, 10. [CrossRef]
2. Guo, B.; Song, S.; Ghalambor, A.; Lin, T. *Offshore Pipelines: Design, Installation, and Maintenance*, 2nd ed.; Gulf Professional Publishing: Oxford, UK, 2014.
3. Ho, M.; El-Borgi, S.; Patil, D.; Song, G. Inspection and Monitoring Systems for Subsea Pipelines: A Review. *Struct. Health Monit.* **2019**, *19*, 606–645. [CrossRef]
4. Korde, U.A.; Ertekin, R.C. Wave Energy Conversion by Controlled Floating and Submerged Cylindrical Buoys. *J. Ocean Eng. Mar. Energy* **2015**, *1*, 255–272. [CrossRef]
5. Leão Shinoka, T.K.; Netto, T.A. Structural Optimization Applied to Submarine Pressure Hulls. *J. Ocean Eng. Mar. Energy* **2025**, *11*, 169–182. [CrossRef]
6. Kolahdouzan, F.; Afzalimehr, H.; Siadatmousavi, S.M.; Yagci, O. Experimental Study on Flow Around Horizontal Multiple Pipelines Laid on the Erodible Seabed Surface. *J. Ocean Eng. Mar. Energy* **2024**, *10*, 555–571. [CrossRef]
7. Hansen, R.C.; Suwandi, A. Visual Inspection of Subsea Pipelines Using ROVs: Capabilities and Limitations. *Mar. Technol. Soc. J.* **2020**, *54*, 14–24.

8. Gašparović, B.; Lerga, J.; Mauša, G.; Ivašić-Kos, M. Deep Learning Approach for Object Detection in Underwater Pipeline Images. *Appl. Artif. Intell.* **2022**, *36*, 2146853. [CrossRef]
9. Zhang, F.; Zhang, W.; Cheng, C.; Hou, X.; Cao, C. Detection of Small Objects in Side-Scan Sonar Images Using an Enhanced YOLOv7-Based Approach. *J. Mar. Sci. Eng.* **2023**, *11*, 2155. [CrossRef]
10. Fiedler, J. An Overview of Pipeline Leak Detection Technologies. In Proceedings of the American School of Gas Measurement Technology, Houston, TX, USA, 22–25 September 2014.
11. Meniconi, S.; Brunone, B.; Tirello, L.; Rubin, A.; Cifrodelli, M.; Capponi, C. Transient tests for checking the Trieste subsea pipeline: Towards the field tests. *J. Mar. Sci. Eng.* **2024**, *12*, 374. [CrossRef]
12. Meniconi, S.; Brunone, B.; Tirello, L.; Rubin, A.; Cifrodelli, M.; Capponi, C. Transient tests for checking the Trieste subsea pipeline: Diving into fault detection. *J. Mar. Sci. Eng.* **2024**, *12*, 391. [CrossRef]
13. Huby, A.A.; Sagban, R.; Alubady, R. Oil Spill Detection Based on Machine Learning and Deep Learning: A Review. In Proceedings of the 2022 5th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, Iraq, 4–5 May 2022; pp. 85–90. [CrossRef]
14. Saleem, F.; Ahmad, Z.; Kim, J.M. Real-Time Pipeline Leak Detection: A Hybrid Deep Learning Approach Using Acoustic Emission Signals. *Appl. Sci.* **2025**, *15*, 185. [CrossRef]
15. Vasconcelos, R.N.; Lima, A.T.C.; Lentini, C.A.D.; Miranda, J.G.V.; de Mendonça, L.F.F.; Lopes, J.M.; Santana, M.M.M.; Cambuí, E.C.B.; Souza, D.T.M.; Costa, D.P.; et al. Deep Learning-Based Approaches for Oil Spill Detection: A Bibliometric Review. *J. Mar. Sci. Eng.* **2023**, *11*, 345. [CrossRef]
16. Li, X.; Li, X.; Han, B.; Wang, S.; Chen, K. Application of EfficientNet and YOLOv5 Model in Submarine Pipeline Inspection and a New Decision-Making System. *Water* **2023**, *15*, 3386. [CrossRef]
17. Wang, W.; Gao, Y. Pipeline Leak Detection Method Based on Acoustic-Pressure Information Fusion and Noise Reduction Algorithm. *Measurement* **2023**, *212*, 112691. [CrossRef]
18. Xu, S.; Zhang, M.; Song, W.; Mei, H.; He, Q.; Liotta, A. A Systematic Review and Analysis of Deep Learning-Based Underwater Object Detection. *Neurocomputing* **2023**, *527*, 204–232. [CrossRef]
19. Xie, Y.; Xiao, Y.; Liu, X.; Liu, G.; Jiang, W.; Qin, J. Time-Frequency Distribution Map-Based Convolutional Neural Network (CNN) Model for Underwater Pipeline Leakage Detection Using Acoustic Signals. *Sensors* **2020**, *20*, 5040. [CrossRef]
20. Chen, P.; Li, R.; Fu, K.; Zhong, Z.; Xie, J.; Wang, J.; Zhu, J. A Cascaded Deep Learning Approach for Detecting Pipeline Defects via Pretrained YOLOv5 and ViT Models Based on MFL Data. *Mech. Syst. Signal Process.* **2024**, *206*, 110919. [CrossRef]
21. Domingos, L.C.F.; Santos, P.E.; Skelton, P.S.M.; Brinkworth, R.S.A.; Sammut, K. A Survey of Underwater Acoustic Data Classification Methods Using Deep Learning for Shoreline Surveillance. *Sensors* **2022**, *22*, 2181. [CrossRef]
22. Moniruzzaman, M.; Islam, S.M.S.; Bennamoun, M.; Lavery, P. Deep Learning on Underwater Marine Object Detection: A Survey. In *Advanced Concepts for Intelligent Vision Systems*; Springer: Cham, Switzerland, 2017; pp. 150–160.
23. Zhang, X.; Shi, J.; Yang, M.; Huang, X.; Usmani, A.S.; Chen, G.; Fu, J.; Huang, J.; Li, J. Real-Time Pipeline Leak Detection and Localization Using an Attention-Based LSTM Approach. *Process Saf. Environ. Prot.* **2023**, *174*, 460–472. [CrossRef]
24. Aubard, M.; Madureira, A.; Teixeira, L.; Pinto, J. Sonar-Based Deep Learning in Underwater Robotics: Overview, Robustness and Challenges. *IEEE J. Ocean. Eng.* **2025**, *50*, 1866–1884. [CrossRef]
25. Malashin, I.; Tynchenko, V.; Nelyub, V.; Borodulin, A.; Gantimurov, A.; Krysko, N.V.; Shchipakov, N.A.; Kozlov, D.M.; Kusyy, A.G.; Martysyuk, D.; et al. Deep Learning Approach for Pitting Corrosion Detection in Gas Pipelines. *Sensors* **2024**, *24*, 3563. [CrossRef] [PubMed]
26. Schøyen, V.S.; Warakagoda, N.D.; Midtgaard, Ø. Seafloor Pipeline Detection with Deep Learning. In Proceedings of the Northern Lights Deep Learning Workshop (NLDL 2021), Tromsø, Norway, 19–21 January 2021.
27. Er, M.J.; Chen, J.; Zhang, Y.; Gao, W. Research Challenges, Recent Advances, and Popular Datasets in Deep Learning-Based Underwater Marine Object Detection: A Review. *Sensors* **2023**, *23*, 1990. [CrossRef]
28. Silva, V. Underwater Pipeline Leak Dataset: Experimental Images from Tank Simulations of Fluid and Gas Releases; Zenodo: Geneva, Switzerland, 2025. [CrossRef]
29. Smart, P.L.; Laidlaw, I.M.S. An Evaluation of Some Fluorescent Dyes for Water Tracing. *Water Resour. Res.* **1977**, *13*, 15–33. [CrossRef]
30. Šegota, S.B.; Lorencin, I.; Anđelić, N.; Mrzljak, V. Use of Synthetic Data in Maritime Applications for the Problem of Steam Turbine Exergy Analysis. *J. Mar. Sci. Eng.* **2023**, *11*, 1595. [CrossRef]
31. Kang, C.; Kim, H.; Lee, J. Machine Learning-Based Anomaly Detection on Seawater Temperature Data with Oversampling. *J. Mar. Sci. Eng.* **2024**, *12*, 980. [CrossRef]
32. Bao, L.; Wang, Y.; Zhang, X. AI-Assisted Inheritance of Qinghua Porcelain Cultural Patterns Using ComfyUI and Stable Diffusion. *Electronics* **2025**, *14*, 725. [CrossRef]
33. Shakhovska, N. High-Fidelity Synthetic Data Generation Framework for Visual Models Based on 3D Representations. *Mathematics* **2025**, *13*, 120. [CrossRef]

34. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597. [CrossRef]
35. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 10684–10695. [CrossRef]
36. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, L.; Wang, W.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685. Available online: https://arxiv.org/abs/2106.09685 (accessed on 22 August 2025).
37. Zhang, L.; Rao, A.; Agrawala, M. Adding Conditional Control to Text-to-Image Diffusion Models. *arXiv* **2023**, arXiv:2302.05543. Available online: https://arxiv.org/abs/2302.05543 (accessed on 22 August 2025).
38. Comfyanonymous; ComfyUI Contributors. *ComfyUI: Node-Based User Interface for Stable Diffusion*. Available online: https://github.com/comfyanonymous/ComfyUI (accessed on 22 August 2025).
39. CivitAI. Realistic Vision V6.0 B1 Checkpoint (realisticVisionV60B1_v51HyperVAE.safetensors). Available online: https://civitai.com/models/4201/realistic-vision-v60-b1?modelVersionId=245598 (accessed on 22 August 2025).
40. CivitAI. deep_ocean_photography_hd LoRA (deep_ocean_photography_hd.safetensors). Available online: https://civitai.com/models/178076/deep-ocean-photography-hd (accessed on 22 August 2025).
41. Xi, W.; Li, Y.; Xu, Z.; Luo, H. Underwater Image Enhancement: A Comprehensive Review and Benchmarking. *Remote Sens.* **2022**, *14*, 4297. [CrossRef]
42. Wang, Y.; Guo, Z.; Chen, J.; Li, C.; Liu, J. Diffusion Models for Underwater Image Enhancement via Self-Supervised Learning. *IEEE Trans. Geosci. Remote Sens.* **2024**, *early access*.
43. Lu, Z.; Liao, L.; Li, C.; Xie, X.; Yuan, H. A Diffusion Model and Knowledge Distillation Framework for Robust Coral Detection in Complex Underwater Environments. *Eng. Appl. Artif. Intell.* **2025**, *159*, 111414. [CrossRef]
44. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed.; O'Reilly Media: Sebastopol, CA, USA, 2019; pp. 259–262.
45. Chollet, F. *Deep Learning with Python*, 2nd ed.; Manning Publications: Shelter Island, NY, USA, 2021.
46. Islam, M.J.; Xia, Y.; Sattar, J. Fast Underwater Image Enhancement for Improved Visual Perception. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3227–3234. [CrossRef]
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. [CrossRef]
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
49. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
50. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
51. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284. [CrossRef]
52. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
53. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
54. Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. In *Python Machine Learning*, 1st ed.; Packt Publishing: Birmingham, UK, 2015; pp. 194–200.
55. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
56. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2018; pp. 270–279.
57. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
58. Deng, X.; Shi, M.; Khan, B.; Choo, Y.H.; Ghaffar, F.; Lim, C.P. A Lightweight CNN Model for UAV-Based Image Classification. *Soft Comput.* **2025**, *29*, 2363–2378. [CrossRef]
59. Wojciuk, B.; Mazur, M.; Wojciuk, P. Improving Classification Accuracy of Fine-Tuned CNN Models. *PLoS ONE* **2024**, *19*, e0289783. [CrossRef]

60. Prechelt, L. Early Stopping—But When? In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 55–69.
61. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
62. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS 2012), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
63. Shorten, C.; Khoshgoftaar, T.M. A Survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
64. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852. [CrossRef]
65. Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M.; Yang, Y.; Zhou, Y. Deep Learning Scaling is Predictable, Empirically. *arXiv* **2017**, arXiv:1712.00409. Available online: https://arxiv.org/abs/1712.00409 (accessed on 10 August 2025). [CrossRef]
66. NOAA Office of Response and Restoration. National Oceanic and Atmospheric Administration (NOAA) Office of Response and Restoration. Available online: https://response.restoration.noaa.gov/ (accessed on 7 October 2021).
67. Skarke, A.; Ruppel, C.; Kodis, M.; Brothers, D.; Lobecker, E. Widespread Methane Leakage from the Sea Floor on the Northern US Atlantic Margin. *Nat. Geosci.* **2014**, *7*, 657–661. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.