Sustainable Vision AI for Oil Spill Prevention

Kuanyin Akech Malang

Chinedu Pascal Ezenkwu

Carlos F. Moreno-Garcia

School of Computing, Engineering and Technology
Robert Gordon University
Aberdeen, Scotland, UK

Abstract

The Oil and Gas industry supplies most of the world's energy, but also poses environmental risks. Specifically, its entire value chain is associated with oil spills that harm ecosystems and coastal livelihoods. This calls for prevention measures as offshore production expands into challenging conditions. Visual surveillance, powered by deep learning models such as Convolutional Neural Networks, U-Net, You Only Look Once, and Reinforcement Learning agents, is widely used to detect leaks and operational anomalies. While these models achieve high accuracy, most studies overlook their energy use, carbon emissions, and lifecycle impacts. This narrative review of research from 2020 to 2025 reveals a clear split between compute-intensive and lightweight designs. The latter delivers nearly the same performance at far lower cost. This highlights the need for lifecycle-aware evaluation, standard efficiency metrics, and design choices that balance accuracy with environmental responsibility. To our knowledge, this is the first review to frame oil spill prevention vision methods explicitly through Green and Sustainable Artificial Intelligence principles.

1 Introduction

The Oil and Gas (O&G) industry still supplies most of global energy even as renewable sources grow, but its operations carry significant environmental risks, such as greenhouse gas (GHG) emissions, pollution, and oil spills [10]. Oil spills in particular damage ecosystems and coastal livelihoods [2]. The urgency of this problem is underscored by recent high-profile spills and the ongoing expansion of offshore operations into riskier environments [15]. Preventing spills requires continuous monitoring of facilities, and visual surveillance through satellite, drone, closed-circuit television (CCTV), and remotely operated vehicle (ROV) acquired imagery, now underpins many systems. Deep learning models, such as Convolutional Neural Networks (CNNs), U-Nets, the You Only Look Once (YOLO) model, and Reinforcement Learning (RL) agents, are applied to detect leaks and anomalies with reported accuracies often exceeding 90% [4, 13, 18, 20, 23, 24]. Despite this progress, most papers focus on accuracy and ignore computational cost and carbon footprint. For example, training large vision models can emit hundreds of tonnes of CO_2 [8, 9, 12, 21]. Such costs are invisible in the current oil spill prevention (OSP) literature: none of the reviewed studies reports energy use, FLOPs, or CO_2 emissions. We highlight this gap based on the Green Artificial Intelligence (AI) and Sustainable AI frameworks, which call for reporting efficiency alongside accuracy [19, 22]. Our goal is to map the energy profiles of OSP vision research and propose guidelines for responsible design and reporting. The following section reviews how different OSP studies approach data, architectures, training, and deployment, and how these choices shape their energy profile.

2 Energy Profile of Vision-Based OSP

2.1 Data Practices and Their Energy Impact

We begin with data practices, as dataset design and augmentation strongly influence both accuracy and energy cost. Some OSP studies expanded small datasets through heavy augmentation or simulation, gaining accuracy at the expense of higher computation. Soares et al. [20] utilised a U-Net for corrosion detection in a storage tank. With only 56 annotated images, they applied extensive augmentations (e.g. blur, lighting, and reflections) to up to 672 images. This raised the segmentation intersection over union (IoU) to $\approx$0.945 but required 500-epoch GPU training (NVIDIA T4), a clearly energy-intensive process, though energy use was not reported. Wang et al. [23] generated synthetic data via a digital-twin ocean model simulating oil spill spread. Thousands of images were used to train a deep Q-network with reinforcement learning to locate spill sources, achieving a simulated accuracy of 98.97%. This involved days of RL rollouts, implying high hidden energy costs. Chagas et al. [3] avoided deep learning, using classical vision (background subtraction, filtering, blob detection) on ROV video to track bubbles and estimate leaks in a subsea oil pipeline. Their C# implementation processed 87 frames per second (fps) on a CPU, far faster and more efficient than prior methods. In summary, heavy augmentation [20] and simulation [23] boost accuracy but increase computational needs, while classical analytics [3] achieve real-time results with minimal energy. Sustainable practice balances these extremes, using augmentation or simulation only when necessary and favouring pre-existing data or analytical methods to avoid redundant training. These cases highlight the tension between accuracy gains and computational overhead, a theme that recurs across model designs.

2.2 Model Architectures and Algorithmic Design

Model design is the most visible source of variation in energy demand, ranging from single CNNs to multi-model ensembles and edge-ready networks. In the following sections, we discuss the most commonly used architectures in terms of their viability to meet efficiency demands.

2.2.1 Single-network CNNs

Ferreira et al. [18] utilised a multi-layer CNN to regress pipeline corrosion defect severity (e.g., predicting remaining wall thickness or burst pressure). The CNN was trained and validated on a combination of finite element method (FEM) simulation data and physical burst test results. The model achieved high accuracy (the authors report an R^2 $\approx$0.95 coefficient, indicating close agreement between predictions and ground-truth). However, the architecture was a fairly standard deep CNN, and the authors did not discuss any efficiency optimisations or measure inference speed. Notably, Ferreira et al. performed extensive validation. Using 100 Monte Carlo cross-validation runs, which, while good for statistical confidence, multiplies the training effort by 100×. This approach yielded a robust model but with a heavy computational burden for training and testing.

Xu et al. [24] proposed a novel architecture called the Immune Depth Presentation CNN (IDPCNN) for oil and gas pipeline fault diagnosis. The key innovation in their work was the addition of specialised 1×1 convolutional layers inspired by biological immune systems to capture inter-feature (channel-wise) correlations, which standard CNNs often overlook. Enhancing CNN with these "immune" layers resulted in the IDPCNN, a five-stage CNN, making it a deeper network with more parameters than baseline models. Upon testing on infrared pipeline images with complex backgrounds, the IDPCNN outperformed classic architectures, such as VGG16, VGG19, ResNet34, and ResNet50, achieving the highest average defect detection accuracy among them. The main trade-off in this work is complexity. The authors note that the enriched network "exhibits a certain level of complexity", which increases computational load and training time. They argue that this is justified for critical fault diagnosis, where accuracy and stability are more important than speed. Xu and Yu's model gains about 2–5% accuracy over simpler CNNs, but at the cost of a larger model with more computation. From a sustainability perspective, it is a compute-intensive design, chosen deliberately to maximise performance in a safety-critical setting.

2.2.2 Ensembles and multi-model pipelines

Dang et al. [4] developed an ensemble of nine deep learning classifiers for event (anomaly) classification in subsea pipeline inspection videos. Their ensemble combined six CNN-based models (VGG16, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, Xception) and three Transformer-based vision models (ViT, MaxViT, Swin-Transformer). Each model was first fine-tuned individually on the datasets, and then their outputs were fused by a weighted voting scheme optimised on a validation set. The result was very high accuracy on the tested datasets: the ensemble achieved around 78% accuracy on two challenging datasets and over 99% on a third dataset with simpler conditions, outperforming any single model in isolation. The accuracy gains, however, came at significant computational cost. Training nine models (even with transfer learning) requires nine times the training computations (the authors used ImageNet-pretrained weights for some models to mitigate this). More importantly, inference with the full ensemble means running all nine models for each new input, which is extremely demanding. Even if each model runs in parallel, the ensemble's inference time is dominated by the slowest network and uses 9× memory. In deployment, such a system would likely need a powerful Graphics Processing Unit (GPU) server to process ROV video frames in real-time. This approach lies at the extreme end of model complexity in our review. It favours high accuracy and robustness (by assembling diverse architectures) over efficiency. This shows that, although ensembles often deliver the highest accuracy, they consume considerable energy and resources, making them unsuitable for low-power or real-time edge use.

Ghorbani et al. [6] implemented a multi-task deep learning pipeline to monitor offshore oil pollution from aerial images. Instead of one unified model, they trained separate models for different subtasks: (1) a VGG16 CNN (with transfer learning) for image-level classification of "oil spill present" vs "no spill", (2) two segmentation networks (Mask R-CNN and PSPNet) for pixel-wise segmentation of the spill area, and (3) a YOLOv3 detector for identifying whether vessels or rigs are in the vicinity of the spill. The rationale was to provide a comprehensive situational picture: detect if a spill is in the image, segment its extent, and detect nearby infrastructure. Each component performed well individually. The VGG16 classifier reached ∼92% accuracy, the segmentation models had a mean IoU of 49% (Mask R-CNN) and 68% (PSPNet) on challenging multi-location data, and YOLOv3 achieved ∼71% mAP in detecting rigs/vessels. However, training three different DL models (even on the same dataset, called "Nafta" with 1292 images) multiplied the cost. Moreover, at inference time, the system must run all three models to fully analyse an image. This is more efficient than Dang et al.'s 9-model ensemble, as each model handles a separate task rather than redundant ensembles on a single task. However, it still exemplifies a compute-heavy design. The authors did not report runtime performance, but a cloud deployment would likely be needed for near-real-time processing of incoming images. There was no mention of model compression or consolidation. It is possible that the system could be implemented as a single multi-head model handling classification, segmentation, and detection to avoid duplicate backbones while training each task from scratch. In essence, the authors presented an ambitious, multifaceted system with strong accuracy, but at the cost of running several large models independently, which leads to substantial training and inference overhead.

2.2.3 Edge-oriented or Lightweight designs

Magana-Mora et al. [13] focused on drilling safety and introduced the "Well Control Space Out" system. It is an IoT and edge computing solution deploying DL models on a drilling rig for real-time visual monitoring. The core task was object detection: spotting the thicker tool joints on the drill pipe in live video, so crews can know exactly where to close the blowout preventer (BOP) during an emergency. The researchers evaluated several modern detector architectures and backbones. Notably, they found that a Faster R-CNN with a ResNet-101 backbone gave the highest precision/recall (best Average Precision, AP) in detecting the relevant objects. However, for deployment, they selected a ResNet-50-based model as the final solution. The ResNet-101 model was slightly more accurate but slower and more challenging to deploy on an edge device. The authors chose the lighter ResNet-50, sacrificing a small amount of accuracy for much faster inference, which translates into a deliberate efficiency trade-off. They also used heavy data augmentation in training and added a custom post-processing step, which increased the detector's AP by about 4.5%, narrowing the gap while maintaining speed. The result was an edge-deployable system that runs in real time (processing video feeds on-site) and improves drilling safety by eliminating human delays that affect oil well control. This approach thus shows a balance of accuracy and efficiency: they started with state-of-the-art models, but then optimised for the edge by selecting a smaller backbone and refining the model to regain accuracy. This contrasts sharply with the approach of using the largest model available. Here, the design was shaped by edge-computing limits, aiming for an architecture that balanced efficiency with accuracy.

Paroha et al. [16] developed a real-time monitoring system for oilfield operations using a shallow, custom deep neural network. The model was built to detect anomalies in multivariate sensor data as they occurred. Given the temporal nature of the data, the architecture included a Long Short-Term Memory (LSTM) layer for time-series inputs, followed by feed-forward layers for classification. It achieved 92.5% accuracy, a detection speed of 0.28 seconds (∼ 4 detections per second), and a "reactivity" score of 96.7%. Training used a broad dataset of sensor readings, operational events, and environmental parameters. The network was optimised for fast inference at the edge, avoiding the need for millions of parameters or ensemble methods. Its rapid runtime suggests it can run on standard industry hardware or edge devices. The design kept complexity just high enough to capture the relevant patterns, delivering over 92% accuracy while minimising energy use. By focusing on only the necessary features and a streamlined architecture, it outperformed traditional non-AI monitoring methods in both speed and accuracy.

2.2.4 Classical computer vision

Chagas et al.[3] (already discussed under data practices in Section **??**, but worth noting here as well for the architecture used) did not use a CNN or any deep model at all, relying instead on classical image processing, which indeed can be considered a fixed algorithmic architecture. Authors reported ∼20–80+ fps throughput on CPU and essentially zero training cost. It is an existence proof that for certain OSP tasks (especially geometric measurements, such as bubble counting), classical methods can meet real-time requirements without the need for heavy AI models. The obvious downside is that such methods may not generalise to complex imagery or multiple conditions compared to a learnable model. But as a design point, it is the most energy-efficient architecture possible: no learned parameters, just efficient code.

Summary: We see two broad categories: energy-intensive architectures – large CNN backbones (VGG, ResNet101), multi-model ensembles, novel complex networks (IDPCNN with extra layers), RL agents requiring iterative simulation – versus energy-efficient architectures – smaller CNNs (ResNet50, YOLO models), shallow bespoke DNNs, and classical CV pipelines. The former often pushes accuracy a few percentage points higher or adds capabilities, but demands much more compute. The latter achieves slightly lower but still high accuracy with a fraction of the resources, and they are viable on edge devices. This contrast highlights the key challenge in OSP vision: delivering the needed accuracy while staying efficient enough to be sustainable and practical to deploy. As we discuss later, techniques such as model compression, knowledge distillation, and multi-task learning can sometimes offer the "best of both" – approaching the accuracy of heavy models with the efficiency of light ones – but these were largely absent in the current literature. In short, accuracy often rises with complexity, but the marginal gains must be weighed against substantial increases in computation.

2.3 Training Strategies and Energy Implications

How a model is trained — including setup, techniques, and iteration loops — can impact the energy required to reach a given accuracy. Our review identified some patterns.

- Augmentation-heavy, long training: Soares et al. [20] trained for 500 epochs on heavily augmented data, a brute-force route to higher IoU. Xu& Yu [24] used a complex staged IDPCNN, likely requiring extensive tuning. Such long schedules can use five times the energy of a 100-epoch run, often for marginal accuracy gains.

- Multi-model ensembles and cross-validation: Dang et al.'s nine-model ensemble [4], Ghorbani & Behzadan's four-model system [6], and Ferreira et al.'s 100-fold CNN training [18] multiplied compute far beyond a single-model baseline. Parallel hardware may hide wall-clock time, but total energy use still scales.

- Simulation-driven reinforcement learning: Wang et al. [23] generated training data on the fly via a high-fidelity ocean simulator, running tens of thousands of RL episodes. Even with transfer learning, the combined simulation and network updates carry a large hidden energy cost. Efficient RL, model simplification, and fewer simulation runs could reduce this burden.

- Repeated validation and hyperparameter search: Extensive testing, as in Wang et al. Monte Carlo [23] or Magana-Mora et al. variant sweeps [13] can dominate total compute. In other AI fields, hundreds of trials are often required for a single result, as is common [19]; OSP risks similar escalation as methods grow more complex.

- Lightweight or no-training approaches: Paroha et al. small DNN [16] trained quickly on modest data; Chagas et al. [3] avoided training entirely, using classical CV on CPUs for real-time results. Such methods cut energy use considerably.

Summary: None of the reviewed studies reported using early stopping, mixed-precision training, batch-size tuning, knowledge distillation, pruning, or quantisation — all common “Green AI” practices [5, 17, 19]. Layer freezing appeared only in Ghorbani et al. classifier [6]. Current OSP training often favours thoroughness over efficiency, with long runs, multiple models, and extra data. Sustainable practice involves reusing pretrained models, stopping when gains plateau, sharing representations, narrowing hyperparameter ranges through small-scale tests, and reporting these details to avoid redundant high-cost experiments. Without such measures, the carbon footprint of “state-of-the-art” remains both large and opaque.

2.4 Deployment Considerations

Deployment context (cloud vs. edge) greatly influences a model’s real-world energy use:

2.4.1 Cloud deployments.

Some methods implicitly assume the use of powerful data centres. Dang et al. 9-model ensemble [4] would likely run on a cloud GPU server for offline ROV analysis. Ghorbani et al. pipeline [6] also suggests central processing of images (e.g., onshore servers). Soares’s U-Net training implies offline GPU use. Cloud deployment concentrates energy in data centres (which may or may not use renewable power) and adds network transmission costs for raw data. The advantage is the ability to run large models and ensembles; the downsides include latency for real-time use and less control over energy sources.

2.4.2 Edge or On-Site Deployments

Several works were designed for resource-limited settings. For example, Magana-Mora et al. [13] deployed their model on a local server at a drilling site, selecting a smaller model for real-time performance on limited hardware. Paroha et al. DNN [16] was intended to run on an industrial PC or microcontroller at the sensor source, enabling a 0.28s response without relying on connectivity. Chagas et al. algorithm [3] was meant for an ROV’s onboard computer, achieving high FPS using only CPUs. Edge deployment inherently enforces efficiency: models must fit memory and power constraints, and raw video can be processed locally to save bandwidth. However, edge hardware is often less energy-efficient per compute unit (no advanced cooling or accelerators), so the models must still be lean.

2.4.3 Hybrid Strategies

A hybrid approach may split training and inference. For example, Wang's RL model [23] can be trained in the cloud (with a high simulation cost) and then deployed on satellite or ground stations for lightweight inference on live images. The training data generation is expensive, but applying the trained model at scale is relatively cheap. Sustainable AI encourages considering this lifecycle: amortise a heavy training cost over many inference uses, and retrain only when necessary. Table 1 summarises edge-targeted studies (Chagas et al. [3], Magana-Mora et al. [13] and Paroha et al. [16]) all using simpler, faster models, and heavier/ensemble networks, mostly cloud-based (Soares et al. [20], Ferreira et al. [18], Xu et al. [24], Dang et al. [4], Ghorbani et al. [6] and Wang et al. [23]). The simpler, faster, edge-targeted studies suggests that "designing with deployment in mind" can lead to more efficient solutions; for example, Dang et al. [4] could have used a single model with multiple outputs instead of an ensemble to save costs.

Table 1: Summary of nine vision-based OSP studies (2020–2025). Efficiency profiles are inferred from model design choices rather than direct energy measurements.

Study(Year)	Approach / Model	Reported Outcome	Profile
Soares et al. (2021)[20]	U-Net CNN with heavy augmentation	IoU ≈0.94 (peak); avg ∼87%	GPU intensive (672 images; 500 epochs; T4 GPU)
Ferreira et al. (2021)[18]	Multi-layer CNN regression on FEM + burst tests	$R^2 \approx 0.95$ for defect severity	High compute demand (100 Monte Carlo CV runs)
Xu et al. (2024)[24]	IDPCNN (immune CNN with 5 stages)	>99% accuracy; better than VGG/ResNet baselines	High complexity; more parameters; long training; not edge-suitable
Dang et al. (2025)[4]	Ensemble of 6 CNNs + 3 Transformers	Accuracy 78–99% across datasets	Very intensive (nine models per input; high inference cost)
Ghorbani et al. (2021)[6]	Multi-task CNNs (VGG16, Mask R-CNN, PSPNet, YOLOv3)	92% classification; IoU 49–68%; mAP ∼71%	High overhead (3 separate networks; cloud-only)
Wang et al. (2022) [23]	Deep Q-Network with transfer learning	98.97% accuracy (Bohai simulations)	Very intensive (days of RL simulation rollouts)
Chagas et al. (2020)[3]	Classical CV pipeline (blob detection)	20–87 FPS on CPU; accurate bubble leak flow estimates	Efficient (no training; real-time CPU execution)
Magana-Mora et al. (2021)[13]	ResNet-50 backbone for drilling safety	∼94–96% detection; +4.5% AP with aug.	Efficient (edge-ready; smaller backbone chosen for latency)
Paroha et al. (2024)[16]	Shallow DNN with LSTM for sensor data	92.5% accuracy; 0.28s detection latency	Moderate (lightweight; realtime on industrial PC)

While Section 2.1 explains how data augmentation and simulation practices affect energy use, Table 2 summarises the dataset characteristics and their implications for the reported outcomes (in Table 1). Most studies rely on small real-world samples (e.g., Soares ∼200 images with only 56 annotated [20]; Xu & Yu 500 IR frames [24]). Others depend on synthetic or simulated data (e.g, Ferreira et al's FEM-based cases [18] and Wang's Bohai oil spill scenario [23]), which may not generalise. Heavy augmentation or simulation is used to compensate for inadequate datasets, inflating accuracy without addressing the underlying scarcity. Even large datasets exhibit imbalance. Dang's ROV Dataset "C" contained only 15 images of concrete damage and 23 of debris, compared with more than 900 exposure and normal images [4]. In all, only Ghorbani & Behzadan [6] provide a moderately sized dataset, while Paroha [16] used proprietary logs, but neither is publicly available. These limitations show that reported accuracies often depend on limited or inaccessible data foundations, emphasising the need to consider dataset quality as part of sustainability-aware evaluation.

Table 2: Dataset characteristics of nine vision-based OSP studies (2020–2025). Most datasets are small, synthetic, or imbalanced, limiting generalisability. Interpreting results requires attention to both dataset quality and unreported energy and carbon costs.

Study (Year)	Dataset	Characteristics
Soares et al. (2023) [20]	∼200 grayscale lab images (only 56 annotated)	U-Net segmentation; binary masks; heavy augmentation (lighting, blur, dust); small, homogeneous; risk of overfitting.
Ferreira et al. (2021)[18]	100 FEM-simulated cases	Synthetic corrosion defects; no augmentation; very small; validated with Monte Carlo CV only.
Xu & Yu (2024)	500 IR pipeline images	Thermal FLIR frames; binary expert labels; augmentation (flips, crops, rotations); balanced but limited scale.
Dang et al. (2024)[4]	ROV frames (A: 819, B: 5212, C: 1891)	Multi-class annotations; extensive augmentation; severe imbalance (e.g., 15 samples for concrete damage).
Ghorbani & Behzadan (2021)[6]	1292 aerial/ocean images	Mixed drone/satellite/ground sources; pixel masks (oil, vessel, rig); balanced spill labels; moderate size.
Wang et al. (2022)[23]	Bohai SAR + simulation	Fully synthetic (Envisat + ECOM model); single-event, scenario-specific.
Chagas et al. (2023)[3]	6 ROV videos (∼14k frames)	Bubble leaks; manual calibration; assumes no overlap; no augmentation; limited coverage.
Magana-Mora et al. (2021)[13]	1100 rig CCTV images	Drillstring tool joints; annotated; heavy augmentation (lighting, color shifts); domain-specific.
Paroha (2024)[16]	Proprietary sensor logs	Pressure, temperature, flow states; extensive preprocessing; no augmentation.

Without sufficiently large, balanced, and representative datasets, training can incur substantial energy and carbon costs without producing deployable solutions. Sustainability-aware evaluation must therefore consider both dataset integrity and transparent reporting of energy and carbon emissions.

3 Sustainability Implications

Across the reviewed studies, accuracy is consistently high (often above 90%), confirming the potential of vision AI in preventing oil spills. Yet sustainability metrics are absent. None reports energy use, carbon emissions, FLOPs, or inference latency (except Paroha, who gives latency indirectly). We contrast "energy-intensive" vs. "energy-efficient" practices below.

3.1 Energy-Intensive Approaches

These maximise performance but at high computational cost:

- Extensive, uncompressed backbones: e.g., Ghorbani et al. [6] VGG16 with ~16M parameters, fine-tuned almost entirely, increasing compute compared to partial freezing.
- Multi-model ensembles: Dang et al. [4] nine-model setup increases the training and inference costs, as well as storage needs.
- Heavy augmentation and long training: Soares et al. [20] 12× augmentation and 500-epoch schedule improved IoU but likely yielded diminishing returns without early stopping.
- Simulation-driven RL: Wang et al. [23] high-accuracy results were achieved through thousands of simulated episodes, each of which was computationally expensive.
- Excessive validation: Repeated training cycles, as in Rezende et al. [18] 100-fold cross-validation, inflate GPU hours. Such practices, while boosting accuracy, risk escalating the field's carbon footprint as data and model sizes grow.

3.2 Energy-Efficient Approaches

These prioritise practicality and lower compute:

- Shallow or compact models: Paroha et al. [16] DNN and Magana-Mora's ResNet-50 detector met accuracy targets (90–96%) and ran on edge devices without GPUs.
- Backbone sharing/multi-task learning [7]: Not used in the reviewed works, but combining tasks in a single network could reduce parameters and computation.
- Classical CV: Chagas et al. [3] blob detection achieved real-time leak tracking on CPUs with negligible energy use.
- Edge-first design: Starting with deployment constraints led to efficient and accurate systems in Paroha et al. [16] and Magana-Mora et al. [13] works.

OSP visual surveillance spans "Red AI" practices that maximise accuracy at high cost and "Green AI" approaches that balance performance with efficiency [19]. The field is still in an early stage, with little attention paid to the cost of achieving high performance. In mainstream AI, this gap is narrowing as tools such as CodeCarbon, CarbonTracker, and

Experiment-Impact-Tracker log energy use and emissions [11]. Publication outlets are also beginning to require efficiency reporting [9]. Despite its environmental focus, the OSP domain has yet to adopt these practices. In some cases, the computation used to prevent spills could itself contribute to emissions, making it essential to design methods that maintain high accuracy while reducing energy demand [14]. Several studies show this is possible: Chagas et al. [3] classical method achieved its goal with negligible carbon cost; Magana-Mora et al. [13] edge-optimised model likely matched heavier systems; Paroha et al. [16] simple DNN outperformed non-AI baselines. These examples suggest many OSP tasks can be efficient by design without losing effectiveness. As Schwartz et al. [19] emphasise, computational efficiency should stand alongside accuracy as a core metric.

4 Recommendations

To align OSP vision research with environmental goals, we recommend:

- Dual performance reporting: Report accuracy alongside efficiency metrics such as FLOPs per inference, compute hours, and estimated energy or CO_2. Simple logging tools can track usage, enabling fair comparisons, and publishers could require this or set carbon limits (e.g., 50 kg CO_2 for train + deploy).

- Efficiency methods: Integrate compression and optimisation during development [17]. Pruning can remove ∼50% of CNN weights with minimal accuracy loss; quantisation (FP16/INT8) speeds inference and cuts energy use; knowledge distillation can replace large ensembles with a single compact model (e.g., distilling Dang et al.'s nine-model ensemble to one CNN retaining ∼95% accuracy). Other options include efficient NAS [5], multi-task learning [7], or even classical ML for small datasets. The bottom line is to treat efficiency as a core design goal.

- Edge-first and TinyML deployment: Many OSP tasks occur in remote, resource-limited settings, so designing for the edge avoids over-engineering and supports solar or battery power. TinyML [1] enables sub-1 MB models on low-power chips, e.g., detecting spills on-device before sending images to save bandwidth. Compact CNNs, such as MobileNets [4] already meet these limits, and an "OSP on the Edge" challenge could spur innovation under strict memory and compute budgets.

5 Conclusion

Visual AI for oil spill prevention achieves high-accuracy detection but often overlooks its environmental cost. Our review of nine studies (2020–2025) found no reporting of energy use or carbon emissions, and a frequent reliance on large models, ensembles, and long training cycles. Yet lighter approaches, including shallow DNNs, edge-optimised networks, and classical CV, delivered similar results with far less computation, showing that accuracy and sustainability can coexist. We recommend lifecycle-aware evaluation, measuring both accuracy and environmental footprint and utilising efficiency-focused methods such as pruning, distillation, and optimised search. Benchmarks should reward models that achieve more with fewer resources. Embedding sustainability into design and evaluation would position OSP research as a model for responsible AI in other climate-sensitive domains.

References

[1] Youssef Abadade, Anas Temouden, Hatim Bamoumen, Nabil Benamar, Yousra Chtouki, and Abdelhakim Senhaji Hafid. A comprehensive survey on tinyml. *IEEE Access*, 11:96892–96922, 2023. ISSN 2169-3536. doi: 10.1109/access.2023.3294111. URL http://dx.doi.org/10.1109/access.2023.3294111.

[2] Mace G. Barron, Deborah N. Vivian, Ron A. Heintz, and Un Hyuk Yim. Long-term ecological impacts from oil spills: Comparison of exxon valdez, hebei spirit, and deepwater horizon. *Environmental Science & Technology*, 54(11):6456–6467, April 2020. ISSN 1520-5851. doi: 10.1021/acs.est.9b05020. URL http://dx.doi.org/10.1021/acs.est.9b05020.

[3] João V. S. Chagas, Gleber T. Texeira, Adriel S. Araujo, André Gonçalves, Fernanda G. O. Passos, and Aura Conci. A computer vision approach to calculate diameter, volume, velocity and flow rate of bubble leaks in offshore wells. In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, page 1–8. IEEE, December 2023. doi: 10.1109/aiccsa59173.2023.10479259. URL http://dx.doi.org/10.1109/aiccsa59173.2023.10479259.

[4] Truong Dang, Tien Thanh Nguyen, Alan Wee-Chung Liew, and Eyad Elyan. Event classification on subsea pipeline inspection data using an ensemble of deep learning classifiers. *Cognitive Computation*, 17(1), November 2024. ISSN 1866-9964. doi: 10.1007/s12559-024-10377-y. URL http://dx.doi.org/10.1007/s12559-024-10377-y.

[5] Nathan C. Frey, Dan Zhao, Simon Axelrod, Michael Jones, David Bestor, Vijay Gadepally, Rafael Gomez-Bombarelli, and Siddharth Samsi. Energy-aware neural architecture selection and hyperparameter optimization. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, page 732–741. IEEE, May 2022. doi: 10.1109/ipdpsw55747.2022.00125. URL http://dx.doi.org/10.1109/ipdpsw55747.2022.00125.

[6] Zahra Ghorbani and Amir H. Behzadan. Monitoring offshore oil pollution using multi-class convolutional neural networks. *Environmental Pollution*, 289:117884, November 2021. ISSN 0269-7491. doi: 10.1016/j.envpol.2021.117884. URL http://dx.doi.org/10.1016/j.envpol.2021.117884.

[7] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International conference on machine learning*, pages 3854–3863. PMLR, 2020.

[8] Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S. Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Chasing carbon: The elusive environmental footprint of computing. *IEEE Micro*, 42(4):37–47, July 2022. ISSN 1937-4143. doi: 10.1109/mm.2022.3163226. URL http://dx.doi.org/10.1109/mm.2022.3163226.

[9] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

[10] Marley Jack. All the evidence against the uk's plans to expand oil and gas drilling. *The Conversation*, August 2023. doi: 10.64628/ab.epex3jxxc. URL http://dx.doi.org/10.64628/ab.epex3jxxc.

[11] Mathilde Jay, Vladimir Ostapenco, Laurent Lefevre, Denis Trystram, Anne-Cécile Orgerie, and Benjamin Fichel. An experimental comparison of software-based power meters: focus on cpu and gpu. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, page 106–118. IEEE, May 2023. doi: 10.1109/ccgrid57682.2023.00020. URL http://dx.doi.org/10.1109/ccgrid57682.2023.00020.

[12] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of machine learning research*, 24(253):1–15, 2023.

[13] Arturo Magana-Mora, Michael Affleck, Mohamad Ibrahim, Greg Makowski, Hitesh Kapoor, William Contreras Otalvora, Musab A. Jamea, Isa S. Umairin, Guodong Zhan, and Chinthaka P. Gooneratne. Well control space out: A deep-learning approach for the optimization of drilling safety operations. *IEEE Access*, 9:76479–76492, 2021. ISSN 2169-3536. doi: 10.1109/access.2021.3082661. URL http://dx.doi.org/10.1109/access.2021.3082661.

[14] Eleanor Mill, Wolfgang Garn, and Nick Ryman-Tubb. Managing sustainability tensions in artificial intelligence: Insights from paradox theory. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, page 491–498. ACM, July 2022. doi: 10.1145/3514094.3534175. URL http://dx.doi.org/10.1145/3514094.3534175.

[15] Ana Cláudia Souza Vidal de Negreiros, Isis Didier Lins, Caio Bezerra Souto Maior, and Márcio José das Chagas Moura. Oil spills characteristics, detection, and recovery methods: A systematic risk-based view. *Journal of Loss Prevention in the Process Industries*, 80:104912, December 2022. ISSN 0950-4230. doi: 10.1016/j.jlp.2022.104912. URL http://dx.doi.org/10.1016/j.jlp.2022.104912.

[16] Abhay Dutt Paroha. Real-time monitoring of oilfield operations with deep neural networks. In *2024 2nd International Conference on Advancement in Computation &amp; Computer Technologies (InCACCT)*, page 176–181. IEEE, May 2024. doi: 10.1109/incacct61598.2024.10551126. URL http://dx.doi.org/10.1109/incacct61598.2024.10551126.

[17] Alvaro Domingo Reguero, Silverio Martinez-Fernandez, and Roberto Verdecchia. Energy-efficient neural network training through runtime layer freezing, model quantisation, and early stopping. *Computer Standards & Interfaces*, 92:103906, March 2025. ISSN 0920-5489. doi: 10.1016/j.csi.2024.103906. URL http://dx.doi.org/10.1016/j.csi.2024.103906.

[18] Guilherme Rezende Bessa Ferreira, Paula Aida Sesini, Luis Paulo Brasil de Souza, Alan Conci Kubrusly, and Helon Vicente Hultmann Ayala. Corrosion-like defect severity estimation in pipelines using convolutional neural networks. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, page 01–07. IEEE, December 2021.

doi: 10.1109/ssci50451.2021.9659884. URL http://dx.doi.org/10.1109/ssci50451.2021.9659884.

[19] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, November 2020. ISSN 1557-7317. doi: 10.1145/3381831. URL http://dx.doi.org/10.1145/3381831.

[20] Luciane Baldassari Soares, Paulo Jefferson Dias De Oliveira Evald, Eduardo Augusto D. Evangelista, Paulo Lilles Jorge Drews-Jr, Silvia Silva Da Costa Botelho, and Rafaela Iovanovichi Machado. An autonomous inspection method for pitting detection using deep learning*. In *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, page 1–6. IEEE, July 2023. doi: 10.1109/indin51400.2023.10218256. URL http://dx.doi.org/10.1109/indin51400.2023.10218256.

[21] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13693–13696, April 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i09.7123. URL http://dx.doi.org/10.1609/aaai.v34i09.7123.

[22] Aimee van Wynsberghe. Sustainable ai: Ai for sustainability and the sustainability of ai. *AI and Ethics*, 1(3):213–218, February 2021. ISSN 2730-5961. doi: 10.1007/s43681-021-00043-6. URL http://dx.doi.org/10.1007/s43681-021-00043-6.

[23] Yuewei Wang, Lizhe Wang, Xiaodao Chen, and Dong Liang. Offshore petroleum leaking source detection method from remote sensing data via deep reinforcement learning with knowledge transfer. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:5826–5840, 2022. ISSN 2151-1535. doi: 10.1109/jstars.2022.3191122. URL http://dx.doi.org/10.1109/jstars.2022.3191122.

[24] Jingyu Xu and Xiao Yu. The immune depth presentation convolutional neural network used for oil and gas pipeline fault diagnosis. *IEEE Access*, 12:163739–163751, 2024. ISSN 2169-3536. doi: 10.1109/access.2024.3358208. URL http://dx.doi.org/10.1109/access.2024.3358208.