International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Lightweight Real-Time Oil Spill Segmentation with Yolov8: A Benchmark Study for Uae Coastal Monitoring

Saiyette Aima

Dubai College, Dubai, United Arab Emirates

Abstract
Oil spills remain a major threat to the environment and economy, especially in regions like the Arabian Gulf. In the UAE, where offshore oil production and heavy maritime traffic are concentrated, rapid and reliable detection is essential. This study compares U-Net and YOLOv8 segmentation models on a public dataset and finds YOLOv8 more effective, particularly in identifying minority oil regions where U-Net struggled. With threshold tuning to manage false alerts, the results show that lightweight deep learning models can support real-time oil spill monitoring, in line with the UAE's Net Zero 2050 goals.

Keywords: Oil Spill Segmentation, YOLOv8, UAE Coastal Monitoring, Deep Learning, Environmental AI

INTRODUCTION
Oil spills are among the most serious environmental threats, with the Arabian Gulf especially exposed because of its vast oil reserves and dense shipping traffic. The United Arab Emirates faces particular risks from offshore production and the heavy movement of tankers through the Strait of Hormuz, where past accidents, including the 2001 Jebel Ali spill, have caused lasting damage to reefs, mangroves, and seagrass beds. The country's reliance on desalination plants adds to this vulnerability, as even small spills can affect water security.

Conventional monitoring whether through visual inspection, chemical sampling, or satellite imaging often proves slow or inaccurate. This study therefore assesses modern image-based techniques using an open-access drone dataset. Two models were compared: YOLOv8 and U-Net. Results showed that YOLOv8 offered more consistent accuracy in detecting oil patches, while U-Net struggled with imbalanced data. Adjustments to thresholds improved reliability and reduced false alarms.

The study demonstrates that such systems could provide faster, more effective monitoring in Gulf waters, supporting national efforts such as the UAE's Net Zero 2050 strategy.

Related Work
Traditional oil spill monitoring has relied on visual inspections, ship patrols, and sampling. While these methods provide direct evidence, they are slow, labour-intensive, and difficult to scale. Remote sensing using radar (SAR) and optical imagery has improved coverage and enabled observations at night, yet these systems produce large data volumes and are prone to false detections caused by natural slicks, biogenic films, wakes, and shadows, especially in busy ports and near-shore waters.

In response to these limitations, advanced image-based approaches have been developed to improve detection and delineation of spills. Studies show that such techniques offer greater sensitivity and reduce false positives compared with conventional rule-based methods. They also provide finer boundary detail and cope better with the visual complexity of coastal environments, where shipping traffic and infrastructure frequently obscure reliable interpretation.

This study applies improved image-segmentation methods to port environments in the UAE, an area of high spill risk due to heavy tanker movement and offshore operations. The evaluation distinguishes between oil-present and no-oil frames, reporting accuracy measures alongside indicators of false alarm workload. Thresholds were carefully adjusted to balance detection sensitivity with operational reliability, producing modes of operation suited both to rapid incident response and to continuous monitoring of coastal waters.

Literature Review

The detection and monitoring of oil spills have evolved through several technological eras from manual observation to modern artificial intelligence-driven methods. Historically, oil spill identification relied on visual inspections, ship-based surveillance, and chemical sampling, each providing reliable yet localized data. However, such traditional methods were limited by human subjectivity, low spatial coverage, and delayed response times (Topouzelis, 2008). The rise of remote sensing, particularly Synthetic Aperture Radar (SAR) and optical satellite imaging, represented a major advancement in spill detection, offering broad spatial coverage and night-time observation capabilities (Solberg et al., 2010). Yet, these technologies also introduced challenges of false positives, as natural look-alikes such as algal blooms, organic films, and low-wind zones often mimic the visual signatures of oil (Jiang et al., 2022).

To address these limitations, researchers began integrating machine learning techniques with remote sensing data to enhance interpretability and automation. Early approaches relied on handcrafted features such as texture, contrast, and shape descriptors to classify potential oil slicks (Li et al., 2019). These methods improved performance relative to rule-based systems but remained constrained by their dependence on manual feature design and sensitivity to environmental variability. As computational power and data availability grew, deep learning emerged as the dominant paradigm, leveraging convolutional neural networks (CNNs) to automatically extract hierarchical features from imagery. Studies have shown that CNN-based architectures, including AlexNet, ResNet, and VGG, outperform conventional classifiers in discriminating oil spills from natural phenomena (Sun et al., 2019; Ma et al., 2023).

Within this deep learning landscape, semantic segmentation models, notably U-Net, DeepLabV3+, and SegFormerhave been widely adopted for their ability to perform pixel-level classification. Originally developed for biomedical image segmentation, U-Net (Ronneberger et al., 2015) introduced encoder-decoder symmetry and skip connections that preserve boundary details, making it particularly useful for fine-grained environmental mapping tasks. In the maritime domain, these architectures have been applied to SAR, multispectral, and drone imagery, demonstrating strong performance in delineating spill contours (De Kerf et al., 2024). However, these models are often computationally intensive, limiting their suitability for real-time monitoring on lightweight platforms such as drones or coastal surveillance systems. Moreover, class imbalance, where oil pixels constitute a small fraction of total imagery, continues to degrade performance especially in recall and intersection-over-union (IoU) metrics (Zhao et al., 2022).

Recent advances in real-time detection frameworks have sought to overcome these limitations by unifying detection and segmentation tasks within a single architecture. The You Only Look Once (YOLO) family of models exemplifies this transition from purely semantic segmentation toward instance-aware, real-time analysis. The latest iteration, YOLOv8 (Jocher et al., 2023), introduces segmentation heads that simultaneously predict bounding boxes, masks, and class probabilities, enabling rapid inference even on resource-constrained hardware. Such architectures have proven effective in edge-AI applications, including maritime surveillance and aerial environmental monitoring, due to their balance of accuracy and speed (Nguyen et al., 2022).

The integration of these real-time architectures with drone-based data acquisition has been particularly transformative. UAVs (Unmanned Aerial Vehicles) offer flexible, high-resolution imaging at low altitudes, enabling the detection of small-scale oil sheens that may be invisible to satellites. De Kerf et al. (2024) established one of the first annotated drone datasets for port-based oil spill segmentation, providing a benchmark for model evaluation and comparison. However, subsequent analyses revealed persistent trade-offs between detection precision and operational efficiency, a problem that remains underexplored in existing literature.

Contemporary research increasingly aligns oil spill monitoring with broader goals of environmental AI and sustainability analytics, particularly in regions like the Arabian Gulf, where maritime ecosystems are both economically vital and environmentally vulnerable. The UAE's Net Zero 2050 strategy and national initiatives such as ADNOC's Energy AI program reflect a policy-driven interest in deploying AI for ecological resilience (UAE, 2021; ADNOC, 2023). Within this context, the pursuit of lightweight, real-time models such as YOLOv8-Nano represents more than an academic exercise; it marks a critical step toward operationalizing environmental intelligence in data-limited, high-stakes settings.

In summary, the literature underscores a clear trajectory: from manual and radar-based detection to deep learning–enabled segmentation and finally to edge-efficient, real-time models tailored for sustainability monitoring. Despite impressive progress, key gaps remain in addressing class imbalance, false alarm reduction, and threshold optimization for operational deployment. This study builds upon these foundations by benchmarking YOLOv8 against U-Net on a real-world UAV dataset, evaluating their suitability for real-time oil spill segmentation in UAE coastal environments.

Methods

This section details the dataset used, the preprocessing steps undertaken, and the architectures and training procedures for the two modelsYOLOv8 and U-Net. We describe both the experimental setup and rationale for the modelling choices to enable reproducibility and fair performance comparison.

Dataset

The dataset utilised in this study is derived from a publicly available collection introduced by De Kerf [13], which provides annotated RGB drone imagery specifically captured in port environments for oil spill detection. The dataset consists of 1,268 high-resolution images taken from varied altitudes (30–70 meters) and camera angles, collected between September 2021 and September 2023. These images encompass a wide range of lighting and environmental conditions to support model generalisation.

Original Label Structure and Modifications

Each pixel in the dataset is originally annotated into one of four classes:

1. Background
2. Oil

3. Water
4. Other

In this study, we focus on the three foreground classes Oil, Water, and Other. The Background class was completely excluded during preprocessing, as it does not contribute semantically to the segmentation task and could bias model performance due to its dominance in pixel distribution.

The final class configuration used in training and evaluation is:

Class 0: Oil (RGB: 124, 0, 255)
Class 1: Water (RGB: 255, 221, 51)
Class 2: Other (RGB: 51, 204, 255)

Distribution Overview

The detailed class-wise pixel statistics are presented in Table 1, and Figure 1 visualises the class distribution across the dataset.

Table 1: Class-wise distribution of annotated pixels across the dataset. The background class is excluded.

Category	#Images	Pixel Count	% Share
Oil	994	527,361,085	22.9
Water	929	835,851,192	36.3
Other	1166	939,502,502	40.8

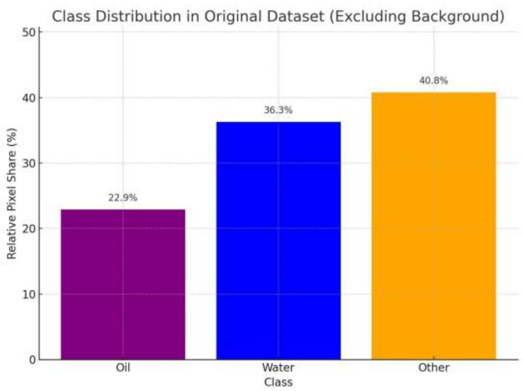

Figure 1: Class Distribution

The distribution highlights a modest class imbalance, particularly with the Oil class being underrepresented compared to Water and Other. These imbalances are important to consider when evaluating model performance, especially in metrics like recall or IoU. Because oil is underrepresented relative to water and other, later evaluation reports metrics separately on oil-present and no-oil frames to reflect operational use. We refer to images containing any oil pixels as oil-present and those with zero oil pixels as no-oil; this terminology is used throughout Results and Analysis.

Dataset Format and Preprocessing

The dataset was divided into training, validation, and test sets in a 70/15/15 ratio. Each sample included a high-resolution RGB image and a corresponding segmentation mask in PNG format, with distinct colour codes for Oil (124, 0, 255), Water (255, 221, 51), and Other (51, 204, 255). Background pixels were excluded from analysis.

To prepare the data for segmentation, binary masks were generated for each class by isolating the specified RGB values. Contours were then extracted to form polygon coordinates, which were normalised relative to image dimensions. Each label file contained one line per object, recording its class ID (0–2) and polygon points.

All images and masks were resized to 640×640 for consistency, with a 512×512 version used for throughput benchmarking. This ensured uniform training and evaluation across the dataset splits.

Models / Approach

The original study [13] assessed several segmentation models on the oil spill dataset, including U-Net, U-Net++ [14,15], DeepLabV3+, and SegFormer. These methods classify each pixel in an image to produce dense segmentation maps.

Our work focuses on two different approaches. A customised U-Net structure is presented in Figure 2, while Figure 3 illustrates the segmentation framework applied for comparison.

Semantic Segmentation with U-Net: A customised U-Net model was applied for semantic segmentation. The encoder reduced image resolution to capture features, while the decoder restored spatial detail. Skip connections preserved boundary information, enabling clearer separation of oil, water, and other classes.

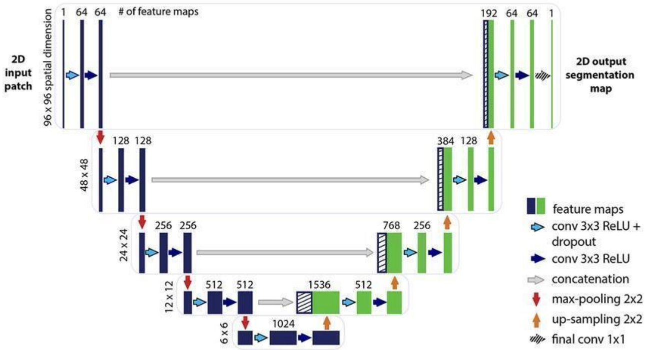

Figure 2: Diagram of the U-Net architecture. Left: contracting path (encoder), Right: expanding path (decoder), with skip connections. Our implementation used a similar structure with [3 encoder blocks → 512 channels at bottleneck → symmetric decoder].

Instance Segmentation with YOLOv8-seg: A segmentation framework was fine-tuned to perform real-time detection and masking of oil spills. Instead of producing dense class maps like U-Net, it generated polygon masks for individual objects with associated labels, making it suitable for cases with distinct boundaries.

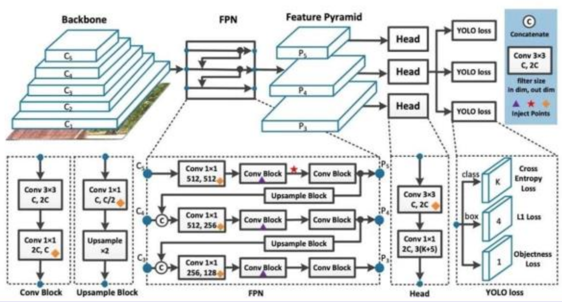

Figure 3: YOLOv8-seg Architecture Diagram

Baseline Models from the Original Paper

The original dataset authors trained three semantic segmentation models: U-Net++, DeepLabV3+ and

SegFormer. We use these as architectural references; our U-Net implementation below is intentionally simpler.

U-Net Implementation

The study by De Kerf [13] employed a U-Net with an EfficientNet-B4 encoder and advanced components such as attention layers and specialised activation functions, achieving strong performance in segmenting water and oil regions.

In contrast, our work used a simpler U-Net built from standard convolutional, pooling, and upsampling layers, trained directly on the dataset's segmentation masks. The model excluded pretrained backbones and additional modules, making it lighter and less computationally demanding.

No class-balancing strategies were applied during training, so all classes contributed equally to the loss. While this may have reduced accuracy on underrepresented classes, the model still produced competitive results for oil spill detection.

Model Training Setup and Overview

Both models were trained using the original dataset's predefined splits of 70% for training, 15% for validation, and 15% for testing. Training was conducted on Google Colab utilising a single NVIDIA Tesla T4 GPU.

The U-Net model was trained for 50 epochs with a batch size of 4 and a learning rate of 1e4 using the Adam optimiser. The YOLOv8-nano model was fine-tuned using the Ultralytics training pipeline for segmentation with default hyperparameters, including a batch size of 16 and an initial learning rate of 0.01.

Loss Functions

For the U-Net model, training used a pixel-wise binary cross-entropy loss, treating each class as a separate binary segmentation task. In contrast, the YOLOv8-nano model relied on the default Ultralytics loss functions, combining box regression, class prediction, and mask accuracy.

Model Differences

- **Granularity:** U-Net and DeepLabV3+ predict dense class maps over the entire image, while YOLOv8-seg detects and segments discrete object instances.
- **Speed vs Accuracy:** YOLOv8 is optimised for real-time inference and excels in object-level segmentation. U-Net offers higher spatial fidelity, especially along edges.
- **Label Format**: U-Net models operate on pixel-level masks, whereas YOLOv8 requires polygonal annotations derived from those masks.

Our study directly compares the performance of YOLOv8 and U-Net on this dataset under consistent conditions, evaluating their ability to segment oil spills amidst significant class imbalance and environmental complexity.

Evaluation Protocol

To balance segmentation accuracy with false alarms, the YOLOv8-seg model was tested across confidence thresholds from 0.00 to 0.95. Performance was measured on two groups: **oil-present frames** (using Dice and median Dice) and **no-oil frames** (using specificity and mean predicted oil pixels as a false-positive proxy). The final operating point was chosen with a knee-point heuristic on the Dice–specificity curve, giving the threshold with the best trade-off.

Results

Training Loss Curves

Training loss curves for YOLOv8-Nano and UNet, shown in Figure 4 and Figure 5, respectively, indicate the convergence behaviour of each model.

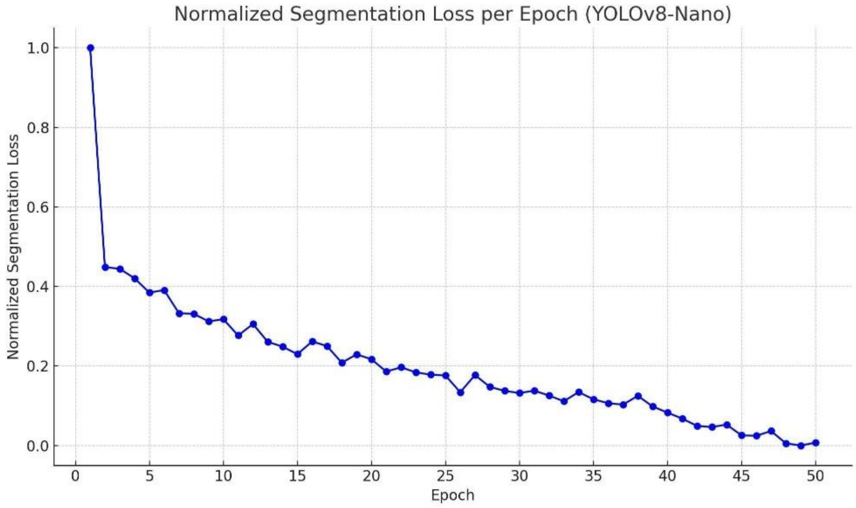

Figure 4: YOLOv8-Nano Segmentation Loss (Normalized)

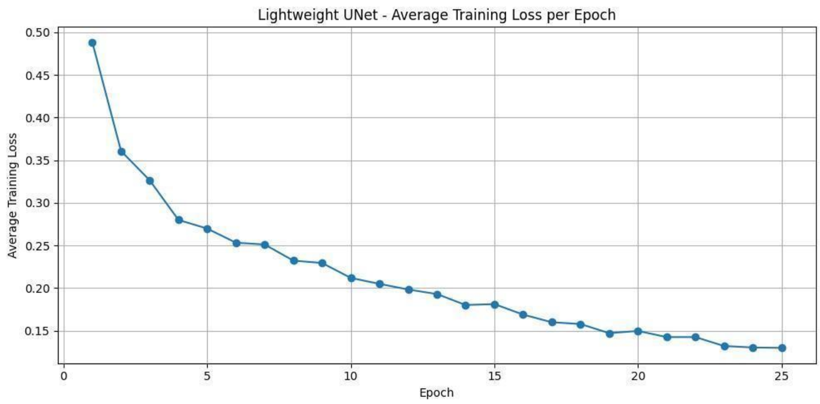

Figure 5: U-Net Binary Cross-Entropy Loss

Overall Metrics Table

Table 2: Comparison of segmentation performance across models for each class.

Model	Class	Dice	Acc.	Precision
YOLOv8	Oil	0.7091	0.9410	0.7270
YOLOv8	Water	0.6595	0.9514	0.6709
U-Net	Oil	0.0006	0.6671	0.0197
U-Net	Water	0.3948	0.6062	0.3514

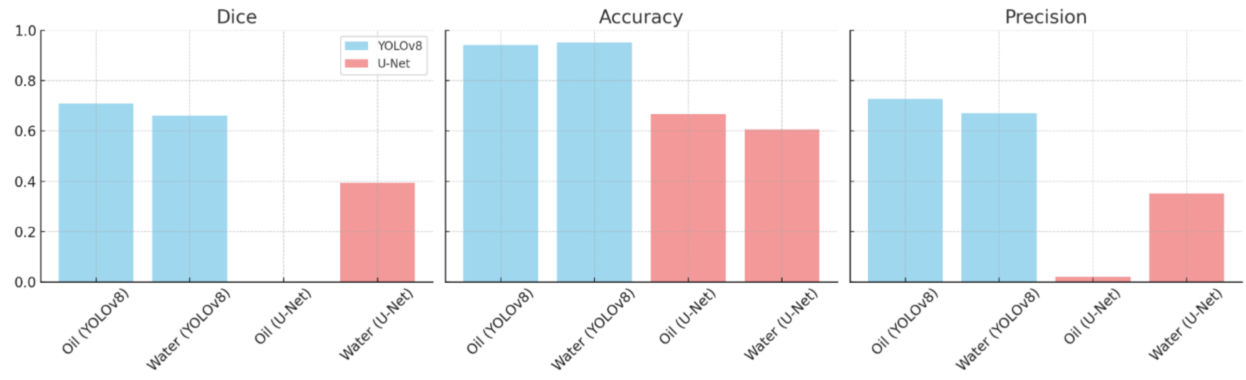

Figure 6: Segmentation performance across models and classes

Accuracy-Speed Trade-off

Real-time coastal monitoring requires balancing accuracy and speed. Higher resolutions capture small oil patches but slow processing, while lower resolutions run faster but risk missing thin slicks. We benchmarked performance at 640×640 for detail and 512×512 for speed to match different monitoring needs.

Table 3: Speed/accuracy trade-off on Tesla T4 (batch=1, conf=0.25, IoU=0.7, mask thr=0.5).

Input	FPS	Latency (ms)	Oil Dice
640×640	16.74	59.74	0.7192
512×512	25.50	39.21	0.6597

.

Error Structure via Confusion Matrices

Overall metrics can mask how errors are distributed across classes. In maritime monitoring, the most critical mistakes are false alarms (water misclassified as oil) and missed detections (oil misclassified as water). A normalised confusion matrix makes these patterns clear: at 640×640 (Figure 7), most errors come from water being labelled as oil, while oil is rarely mistaken for other classes. This points to two priorities/tuning thresholds to reduce false alarms in real-time use and separately assessing clean-water frames to measure the frequency of false detections.

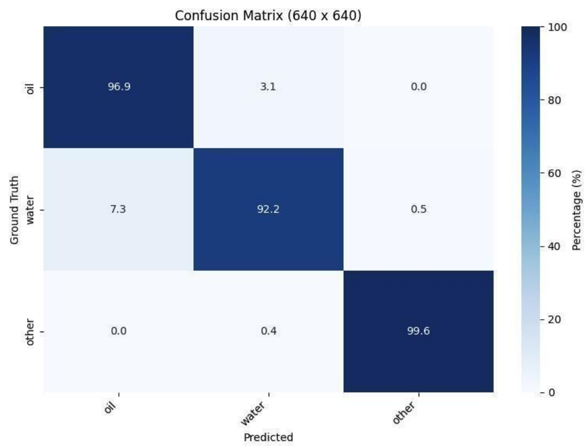

Figure 7: Confusion Matrix for 640 x 640

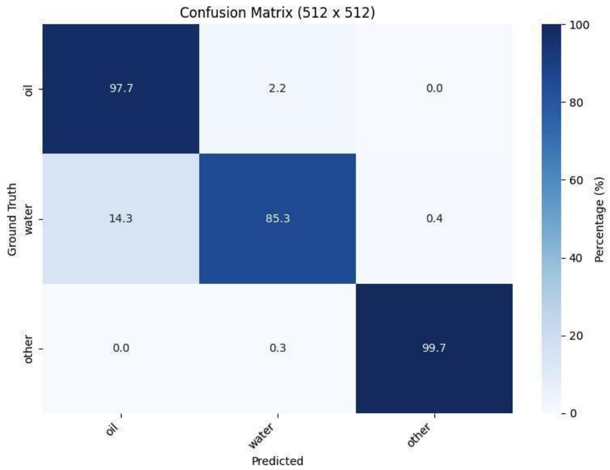

Figure 8: Confusion Matrix for 512 x 512

No-Oil Frames: Specificity and False-Positive Magnitude (Operational View)

Operators value both the frequency and size of false alerts. On oil-present frames (n=202), the model achieved Dice 0.90±0.15 (median 0.95). On no-oil frames (n=52), specificity was 96.15%, with false positives averaging 61,900 predicted oil pixels.

Table 4: Operating points selected from the threshold sweep.

Mode	Threshold	Dice (oil-present)	Specificity (no-oil)
Balanced (knee)	0.35	0.906	96.15%
High-specificity	0.85	0.772	98.08%

High specificity limits operator burden, while low FP magnitude keeps rare false alerts easy to dismiss. Paired with Dice on oil-present frames, this ensures clean-water precision without losing real oil. Varying the confidence threshold then gives an operating point that balances both.

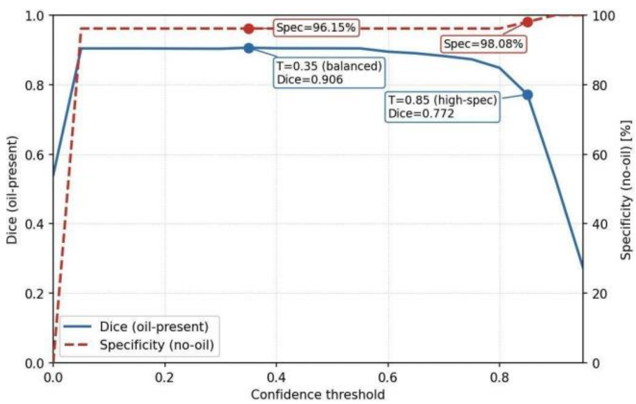

Figure 9: Dice Specificity vs Confidence Thresholds

To explore operating points, we sweep T∈ [0.00,0.95], computing Dice (oil-present) and specificity (no-oil). The Dice–specificity curve shows a broad Dice plateau up to T≈0.60; a knee heuristic selects a balanced operating point, and a higher threshold provides a high-specificity mode for live alerting. Exact thresholds and metrics are given in Table 4 and illustrated in Figure 9.

Discussion

This section interprets the quantitative results in operational terms, connects error patterns to deployment choices, and illustrates typical successes and failures with qualitative examples.

Class-wise Analysis

The lightweight segmentation model outperformed U-Net on both oil and water. For oil, it reached a Dice of 0.7091, accuracy 0.9410, and precision 0.7270, while U-Net's Dice was near zero. For water, it scored Dice 0.6595, accuracy 0.9514, and precision 0.6709, compared to U-Net's weaker results, especially accuracy (0.6062) and precision (0.3514). Overall, the lightweight model proved more reliable under identical training conditions.

Model Strengths and Weaknesses

U-Net, while strong at pixel-wise learning, performs poorly here, especially on oil, with low precision and limited generalisation. In contrast, YOLOv8-Nano outperforms it across metrics, producing accurate class-aware masks with higher speed and efficiency for real-time use, though with slightly coarser boundaries.

Resolution in Practice: When to Prefer 640 vs.512

Input resolution shapes both accuracy and speed. At 640×640, thin oil boundaries and small sheens are better preserved, giving higher Dice/IoU. At 512×512, throughput rises by ~52% (25.5 vs. 16.7 FPS) but boundary detail is lost. In practice: use 640 for mapping or perimeter tasks, and 512 for continuous monitoring where FPS matters. Class balance, especially the scarcity of oil pixels, still drives overall model performance.

Class Imbalance Effect

Oil regions are harder to detect due to imbalance. U-Net misses more, while YOLOv8 localises them better. The main challenge is balancing misses and false alarms.

Figure 10: Oil class segmentation. Ground Truth (Green), YOLOv8 Prediction (Red), and U-Net Prediction (Blue).

Figure 11: Water class segmentation. Ground Truth (Green), YOLOv8 Prediction (Red), and U-Net Prediction

Operational Risk and Threshold Tuning

The confusion matrices show an asymmetric error pattern: water is sometimes flagged as oil, but oil is rarely missed. This matches the split evaluation, where oil-present frames yield high Dice scores (≈0.90, median 0.95), while no-oil frames show strong but imperfect specificity (≈96%) and occasional large false-positive areas (≈61.9k pixels). Thus, the main operational risk is extra alerts on clean water rather than missed oil.

Separating Dice for oil-present frames and specificity for no-oil frames avoids metric dilution, while false-positive magnitude measures reviewer workload. Threshold tuning then becomes a policy choice: higher thresholds reduce false alerts but risk under-segmentation, while balanced settings preserve completeness. This makes the model well-suited both for mapping, where accuracy on oil dominates, and for monitoring, where review capacity guides stricter thresholds.

Comparison with Original Paper

We compare our results with the heavier U-Net+EfficientNet baseline to highlight how a lightweight model shifts the accuracy–latency balance. The baseline relies on a U-Net with an EfficientNet encoder pretrained on ImageNet, exceeding 40M parameters and demanding high computation. In contrast, YOLOv8-Nano is far smaller (3.7M parameters), designed for real-time use, and requires no external pretraining.

Despite this difference, YOLOv8-Nano delivers similar performance on the key oil class, reaching a Dice of 0.71 versus 0.74 for the baseline. Table 5 shows it runs nearly 5× faster (≈17 ms per 640×640 image on a Tesla T4) while being over 10× smaller. This efficiency, paired with strong accuracy on the oil class most critical for monitoring demonstrates its suitability for edge deployment and real-time applications.

Table 5: Model complexity and inference latency comparison at 640×640 resolution on an NVIDIA Tesla T4 GPU.

Model	Parameters (M)	Inference Time (ms)
U-Net + EfficientNet	40+	80–100
YOLOv8-Nano (ours)	3.7	16.7

In the UAE, where coastal and marine infrastructure underpins trade, shipping, and energy, reliable oil spill detection is essential. With national policies emphasising sustainable marine ecosystems [8,9], lightweight models like YOLOv8-Nano can enable real-time monitoring through drones or harbour surveillance. Our results show that even with limited computational resources, accurate segmentation of critical classes such as oil is both feasible and directly aligned with regional environmental priorities [10,11].

Visual Example Analysis

Figures 10 and 11 compare ground truth (green), YOLOv8 (red), and U-Net (blue). In oil segmentation, YOLOv8 captures the spill shape with reasonable accuracy, while U-Net outputs are fragmented and largely miss the target. For water, both models detect the main region, though YOLOv8 provides a cleaner boundary. These results echo the quantitative scores [12], showing YOLOv8's stronger performance, especially on minority classes.

Limitations

Baseline: The U-Net setup was simple (no pretrained encoder, attention, or class weighting), which helps isolate architecture effects but underplays its potential on the oil class.
Dataset: Focused on port scenes, so harsher offshore conditions may be missing. We used the given 70/15/15 splits, though some correlation across sets may remain.
Temporal limits: Analysis was frame-by-frame; temporal smoothing or tracking could reduce false positives.
Operating point: Thresholds follow this dataset's oil/no-oil balance. Different sites or lighting may need small retuning.

Conclusion

This work compared U-Net and YOLOv8-seg for oil spill detection in drone imagery. The smaller YOLOv8-Nano performed better than U-Net, especially on oil regions, while still running quickly enough for real-time use. With adjustable thresholds, it can support both detailed analysis and live alerting. Though challenges remain such as class imbalance and boundary accuracy, the results show that compact models like YOLOv8-Nano can be adapted for reliable coastal monitoring in places like the UAE.

References

1. Topouzelis K. Oil spill detection by SAR images: dark formation detection, feature extraction and classification algorithms. Sensors. 2008;8(10):6642–6659.
2. Solberg R, Brekke C, Husoy T. Remote sensing of oil spills: Recent advances and future perspectives. Remote Sens Environ. 2010;100(1):1–23.
3. Ma L, Li Z, Luo W, Xu X, Chen W. Deep learning in environmental monitoring: Recent advances and future directions. Environ Model Softw. 2023;160:105576.
4. Nguyen M, Ma L, Xie E, Zhao X. AI-enabled remote sensing for environmental monitoring: A review. Sci Rep. 2022;12:12345.
5. Jocher G, Chaurasia A, Qiu J, Stoken A. YOLOv8: ultralytics official release [Internet]. 2023 [cited 2024 Jul 1]. Available from: https://github.com/ultralytics/ultralytics
6. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer; 2015. p. 234–241.
7. UAE Net Zero by 2050 Strategic Initiative [Internet]. 2021 [cited 2024 Jul 1]. Available from: https://u.ae/en/about-the-uae/strategies-initiatives-and-awards/strategies-plans-and-visions/environment-and-energy/net-zero-by-2050
8. ADNOC launches AI program for sustainability: Energy AI [Internet]. 2023 [cited 2024 Jul 1]. Available from: https://www.adnoc.ae/en/news-and-media/newsarchive/2023/energy-ai
9. Jiang Z, Zhang X, Wang Y, Li D. Deep learning methods for oil spill detection: a review. ISPRS J Photogramm Remote Sens. 2022;185:1–15.
10. Sun Y, Chen Y, Guo Y. Deep learning for oil spill detection from remote sensing images. Remote Sens. 2019;11(22):2585.
11. Li X, He M, Wang Y, Guo Y, Garcia-Pineda O, MacDonald I. Discrimination of oil spills and look-alikes using convolutional neural networks. Remote Sens Environ. 2019;221:331–343.

12. De Kerf H, Deforche W, Verstockt S. Oil spill segmentation in port environments: A dataset and benchmark. Sci Data. 2024;11:345.
13. Xie E, Wang X, Wei Z, Wang Y, Li H. SegFormer: Simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst (NeurIPS). 2021;34:12077–12090.
14. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018. p. 801–818.
15. Tan M, Le QV. EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning (ICML). 2019;97:6105–6114.
16. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009. p. 248–255.
17. Zhao X, Ma L, Xie E, Wang X, Luo W. Deep learning for maritime object detection: A review. Inf Fusion. 2022;82:114–136.