Oil Spill Detection on SAR Images
A Comparative Analysis of Five Segmentation Models

Bianca Bartoli
s322799@studenti.polito.it
Politecnico di Torino
Torino, Italy

Carlo Marra
s334220@studenti.polito.it
Politecnico di Torino
Torino, Italy

Alessandro Valenti
s328131@studenti.polito.it
Politecnico di Torino
Torino, Italy

Abstract

The proposed study investigates the application of semantic segmentation models for oil spill detection using Synthetic Aperture Radar (SAR) imagery. The performance of five models, namely U-Net, LinkNet, PSPNet, DeepLabv3+, and CBD-Net, is evaluated on a dataset from the Copernicus Sentinel-1 mission. DeepLabv3+ demonstrates the highest mean Intersection over Union (IoU), offering a balance of accuracy and computational efficiency, while U-Net and LinkNet also perform well in detecting oil spills and small objects. PSPNet and CBD-Net display worse performances compared to the other models, particularly struggling with underrepresented classes in the dataset. Additional tailored techniques, aimed at enhancing generalization and accuracy, are performed to make the network focus on oil spills and improve generalization on this class. The results obtained highlight the potential of deep learning in SAR-based oil spill detection, emphasizing the need for improved noise reduction, hybrid architectures, and lightweight models for real-time applications in satellite monitoring.

1 Introduction

Marine oil pollution caused by human activities poses a significant threat to ecosystems, endangering marine biodiversity and environmental stability. Major oil spills are primarily attributed to accidents at offshore oil drilling platforms, industrial activities, or the illegal discharge of oily residues from ships. The prompt detection of oil slicks, enabling immediate notification of relevant authorities capable of responding and mitigating the threat, is essential to safeguarding both human populations and marine ecosystems.

The rapid detection of oil spills is facilitated by remote sensing technology, which enables efficient and timely monitoring of marine pollution events. Synthetic Aperture Radar (SAR) sensors, deployed on aircraft or satellites such as the Copernicus Sentinel-1 mission by the European Space Agency (ESA), have become a cornerstone technology for oil spill detection due to their reliability and advanced capabilities.

A key advantage of SAR technology is its ability to operate under all weather and lighting conditions, as microwaves can penetrate clouds, smoke, and darkness. Environmental factors such as wind conditions or the presence of algae can interfere with accurate detection by producing dark areas in the imagery, commonly referred to as look-alikes, which complicate precise classification and differentiation from actual oil spills[1–4].

The application of neural networks presents an effective solution for extracting relevant features and performing classification, thereby facilitating precise semantic segmentation of oil spill areas. Specifically, deep learning models designed for semantic segmentation enable the accurate identification of multiple classes within SAR imagery.

In this study, an assessment of the performance of state-of-the-art semantic segmentation models is developed, including U-Net, PSPNet, LinkNet, and DeepLabV3, for the task of oil spill detection. Furthermore, the integration of the Contextual and Boundary-Supervised Detection Network (CBD-Net) is proposed to enhance segmentation accuracy. The implementation of tailored training techniques is also introduced as an attempt to improve the robustness and generalization capabilities of the models.

The remainder of this paper is structured as follows: Section 2 provides an overview of related works on semantic segmentation architectures and their applications on SAR images. Section 3 outlines the methodology and configurations. Section 4 presents and analyses the results. Finally, Section 5 summarizes the findings and introduces potential directions for future research.

2 Related Works

Despite semantic segmentation [5, 6] has proven itself to be a powerful tool for detecting oil spills in SAR images, one of the main challenges remains the scarcity of labeled SAR datasets since they are costly and often not openly accessible.

2.1 Dataset

For this research project, all models are trained using the dataset created by Krestenitis et al. [7]. This dataset consists of satellite SAR images depicting oil-polluted sea areas, accompanied by corresponding labeled masks. The dataset includes five distinct classes: Sea Surface, Oil Spill, Look-Alike, Ship, and Land. The underlying data used to produce these images are sourced from the European Space Agency (ESA) database through the Copernicus Sentinel-1 mission.

The dataset consists of a total of 1,112 images, each with a resolution of 1,250 × 650 pixels. The images are divided into two sets: 90% for training (1,002 images) and 10% for testing (110 images). For each image, two distinct masks are provided. The first mask consists of three RGB channels, representing the five different classes, while the second mask is a single-channel representation, where each pixel contains the label corresponding to its assigned class: 0 for Sea Surface, 1 for Oil Spill, 2 for Look-Alike, 3 for Ship, and 4 for Land. Conversely, the single-channel label mask is utilized for training and evaluation purposes. This label mask, along with the original SAR image, is used as target for the models.

2.2 Models

Convolutional neural networks (CNNs) have become the state-of-the-art approach for this task due to their ability to learn complex spatial and contextual features[8, 9]. These models typically adopt an encoder-decoder architecture. The encoder extracts high-level semantic features from the input image while progressively reducing its spatial resolution, capturing abstract and global information. In contrast, the decoder reconstructs a pixel-wise prediction map from the encoded features, incrementally restoring spatial resolution to match the input size. This architecture enables the integration of global context with local details, improving segmentation accuracy. Different semantic segmentation models differ in their architectural designs, feature extraction methodologies, and trade-offs between computational efficiency and performance. For this project, all the models are implemented with a ResNet-101 encorder backbone, except for DeepLabv3+, which features a lighter architecture and is implemented with MobileNet-100 encoder. The subsequent sections provide a detailed analysis of each of the models utilized in this study.

2.2.1 UNet

U-Net architecture[10] has an "U"-shaped structure and consists of two main components: a contracting path (encoder) and an expansive path (decoder).
The contracting path reduces the spatial dimensions of the input image and increases the number of feature channels, through the repetition of two 3×3 convolutions with ReLU activation function and a 2×2 max-pooling operation.
Instead, the expansive path restores the spatial resolution by upsampling the feature maps using transposed convolutions, followed by concatenation with the corresponding high-resolution feature map from the encoder through skip connections. The connections between the two paths aim to capture high-level features and details. The concatenated feature map is refined with two additional 3×3 convolutions, progressively reducing the number of feature channels. The output of the network is generated by a 1×1 convolution to the feature map at the last decoder layer, which assign each pixel to the corresponding class through a Softmax fuction.

2.2.2 LinkNet

Similarly to UNet, LinkNet[11] consists of three main components: an encoder, a decoder, and a linking mechanism.
The encoder is based on a residual network (ResNet) backbone, where residual blocks of each layer extract hierarchical features. The encoder progressively reduces spatial dimensions using stride-2 convolutions, halving the resolution at each step.
The linking mechanism between the encoder and decoder transforms the extracted features into a form suitable for reconstruction process applied by the decoder. Each decoder stage consists of two main steps: upsampling and residual refinement. Skip connections directly link encoder features to corresponding decoder layers, skipping intermediate layers to preserve spatial details.
The output is produced by a 1x1 convolution and a Softmax for the classification of the pixels.

2.2.3 PSPNet

PSPNet (Pyramid Scene Parsing Network)[12] special feature is the pyramid pooling module incorporating a strategy to aggregate multi-scale contextual information.
The architecture of PSPNet consists of three main components: a feature extraction backbone, a pyramid pooling module, and a segmentation head. The backbone is usually a ResNet and it extracts hierarchical feature maps from the input image through convolutions. The feature maps capture high-level semantic information, but lack in understanding the global context.
Then, the feature map is passed through four parallel pooling layers with different spatial bin sizes ($n \times n$), where $n \in \{1, 2, 3, 6\}$. Each of them applies a global average pooling operation over non-overlapping regions of the feature map, reducing its resolution while preserving global information. The pooled maps are upsampled to the original resolution using bilinear interpolation and concatenated with the original backbone feature map to produce a multi-scale feature representation.
The concatenated feature map is then passed through a 1×1 convolution to reduce the channel dimensions and fuse the multi-scale features. The pixel labeling is produced with a final series of convolutions.

2.2.4 DeepLab

DeepLab[13] is a series of architectures that focus on multi-scale context and precise object boundary detection. Significant innovations integrated in DeepLab architecture are the atrous (dilated) convolutions and Atrous Spatial Pyramid Pooling (ASPP). Atrous convolutions control the effective receptive field without reducing spatial resolution introducing a dilation rate, which spaces out the convolutional kernel, allowing the network to capture global context efficiently.
The version used in this project is DeepLabv3+, the most advanced version of the DeepLab series, which added a lightweight yet effective decoder module for precise boundary refinement. It has three main components: the backbone, the ASPP module, and the decoder.
Atrous convolutions are used in later layers of the backbone to expand the receptive field without reducing spatial resolution, allowing the model to capture both global context and more detailed features. Then, ASPP processes the feature map from the backbone through multiple parallel branches composed of Atrous convolutions with different dilation rates for capturing features at varying scales. Global Average Pooling aggregates image-wide context, followed by a 1×1 convolution. The outputs are concatenated and passed through a 1×1 convolution with batch normalization and activation to fuse multi-scale features.
At the end of the network, the decoder refines the output from ASPP to recover fine spatial details, particularly at object boundaries.
The benefits of DeepLabv3+ are effectiveness for complex segmentation tasks and computational efficiency.

2.2.5 CBD-Net

CBD-Net [14] (Contextual and Boundary-supervised Detection Network) is a convolutional neural network that addresses challenges of precise boundary localization and effective feature extraction by integrating boundary attention mechanisms. The inclusion of novel architectures like CBD-Net has further advanced the field by simultaneously capturing contextual information and enhancing edge detection. The architecture comprises an encoder-decoder structure with skip connections and an scSE attention blocks and

multi-parallel dilated convolutions.
The encoder extracts hierarchical features using residual blocks. Each residual block consists of two convolutional layers with batch normalization and activation, followed by a skip connection that adds the input directly to the output. The encoder processes the input image capturing high-level contextual features while reducing spatial resolution. After each stage, an scSE attention block is applied.
The scSE block enhances the network's ability to focus on important regions by learning both spatial and channel-wise attention.
The decoder reconstructs the spatial resolution of the feature maps while refining boundary details. The decoder process is done by upsampling the feature map through transposed convolutions, then it is added to the corresponding encoder feature map via skip connections. Multi-parallel dilated convolutions are applied at certain stages to capture multi-scale context.

3 Method

The class distribution within the dataset is significantly imbalanced, as reflected by the total pixel counts for each class. The Sea Surface class dominates with 797.7 million pixels, while the Oil Spill and Look-alike classes have substantially fewer, at 9.1 million and 50.4 million pixels, respectively. The Ship class is the most underrepresented, comprising only 0.3 million pixels, whereas the Land class includes 45.7 million pixels. This pronounced imbalance, with the Sea Surface class vastly outweighing the others and the Ship class being particularly sparse, poses a challenge during the training process. Underrepresented classes are more difficult for the network to predict accurately.

3.1 Pre-processing

To enhance the diversity and variability of the dataset, data augmentation techniques are applied to the training set during each epoch of the training phase. Initially, a random resize with a scale factor ranging from 0.5 to 1.5 is applied to 50% of the images. Additionally, 50% of the images are randomly flipped—vertically, horizontally, or both. Finally, a random crop is performed on all images, and a patch of size 320 × 320 pixels is extracted and fed into the models.

3.1.1 Possible data augmentation extensions

To further increase the variability of the dataset and address the underrepresentation of pixels corresponding to oil spills, two additional training methods are implemented.

First, a customized focused crop technique is applied to 50% of the training images, with a higher probability of selecting regions containing oil spills. This approach ensures that the model is exposed to a greater number of samples featuring oil spill pixels, facilitating convergence on the most relevant class for the task.

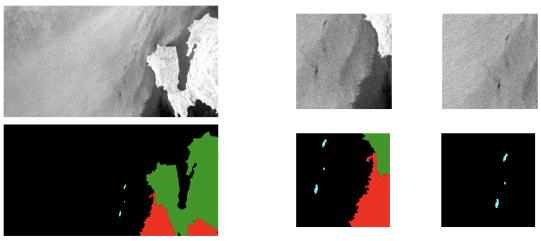

Figure 1: Examples of the focused crop technique. The original image (left) is cropped into 320×320 regions (right) with a 50% probability of including oil spills. Random vertical and horizontal offsets are applied to prevent overfitting to fixed positions, ensuring a diverse training dataset.

To prevent positional bias, a random horizontal and vertical offset, uniformly sampled between 0 and 80 pixels, is added to the crop location. This ensures that oil spills are not consistently centered within the 320×320 cropped regions but instead appear in varied positions, such as towards the edges or corners. By introducing this variability, the network is encouraged to learn robust features without relying on a fixed spatial relationship between the crop and the oil spill. After cropping, images are further augmented with random horizontal and vertical flips, as detailed in the introduction of Section 3.1, further increasing diversity in the training set. An example is illustrated in Figure1.

The second training variation involves applying a cropping augmentation using a sliding window mechanism. The sliding window systematically moves across the image, extracting patches of size 320 × 320 pixels to ensure complete coverage of the image's surface, as Figure 2 shows. This approach ensures that all regions of the image are represented in the dataset, thereby ensuring the diversity of the training data.

Given that the original image dimensions are 1250 × 650 pixels, a patch size of 320 × 320 can not fully cover the entire image surface. To address this, 5 rows of pixels are removed from both the top and bottom of the image, and a reflective padding of 15 pixels is added to both the left and right sides. This adjustment results in a resized image with dimensions of 1280 × 640 pixels, which can be perfectly divided into a 4 × 2 grid of patches, ensuring complete coverage.

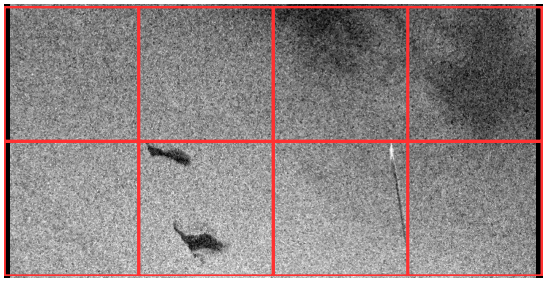

Figure 2: Example of sliding window approach to extract 8 patches with size 320 × 320

3.2 Technical setup and configurations

For all the networks evaluated, no pre-trained weights are used. This choice is made because most pre-training processes rely on ImageNet, a large dataset of RGB images, which differs significantly in nature from SAR images.

The original training set, comprising 1,002 images, is divided into 85% for training and 15% for validation to ensure a robust evaluation of model performance during training.

Each model is trained for a total of 600 epochs, utilizing Cross Entropy as the loss function. To address the class imbalance inherent in the dataset's pixel distribution, custom weight adjustments are applied to the loss function. These weights are maintained consistently across all models to ensure comparability of results. The Adam optimizer is employed for all models to facilitate efficient gradient-based optimization. Instead, the learning rate configuration is fine-tuned for each model.

The chosen metric for evaluation is the Intersection over Union (IoU), which provides a robust measure of segmentation accuracy by calculating the overlap between the predicted and ground truth regions relative to their union. It is computed as:

$$\text{IoU} = \frac{\text{prediction} \cap \text{ground truth}}{\text{prediction} \cup \text{ground truth}} = \frac{\text{TP}}{\text{FP} + \text{TP} + \text{FN}}$$

All models are implemented using PyTorch Lightning, a high-level framework built on PyTorch that simplifies and organizes the training process. By abstracting repetitive tasks such as GPU acceleration, checkpointing, and logging, it allows for a more structured and maintainable codebase.

The training of the models is conducted using an NVIDIA GeForce RTX 4080 Super GPU with 16 GB of VRAM. Batch sizes are carefully optimized to maximize the use of available memory. Table 1 reports the size of the models, their parameter count, the batch sizes used, and the memory occupation on the GPU during training. CBDNet required more computational resources and is trained on an NVIDIA A100 GPU, which provides 40 GB of VRAM.

Model	Size (MB)	Parameters	Batch	Memory (GB)
UNet	541.67	51.5M	20	14.2
LinkNet	539.74	50.2M	20	14.2
PSPNet	168.76	43.3M	32	15.2
DeepLabv3+	49.88	4.7M	36	14.4
CBD-Net	409.99	38.8M	8	22.0

Table 1: Comparison of Model size, Number of parameters, Batch size, and Memory allocation

4 Experiments and evaluation

4.1 Results

Figure 3 shows the evolution of the validation mean IoU during training across the five models. PSPNet being the fastest to reach convergence and DeepLabV3+ consistently scoring the highest.

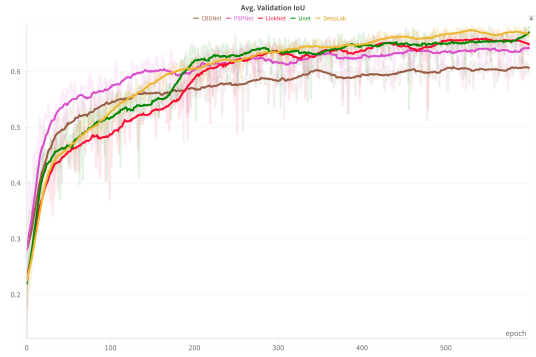

Figure 3: IoU on the validation set during training for the five models.

The models are finally tested on the test set for evaluation, and results are reported in Table 2.

All the models performe similarly on each class, underscoring the difficulty of achieving accurate segmentation on SAR images. In particular, all the models struggle to accurately segment ships out of the images, which is an expected result when considering that it is the class with the lowest representation in the dataset.

4.2 Quantitative analysis

Table 2 summarizes the per-class and mean IoU performance of U-Net, LinkNet, PSPNet, DeepLabv3+, and CBD-Net. Among these, DeepLabv3+ achieves the highest mean IoU (0.6454), closely followed by U-Net (0.6440) and LinkNet (0.6398). U-Net shows the strongest performance in the "Oil Spill" class (0.5554 IoU), slightly outperforming the other models. It also performs well in the "Land" class (0.9500 IoU). LinkNet, on the other hand, demonstrates the best results in detecting the "Ships" class (0.3405 IoU), suggesting its strength in identifying small objects, even though its overall mean IoU is slightly lower.

The performance of U-Net and LinkNet is particularly noteworthy given their simpler architectures. These models do not employ advanced modules, such as atrous convolutions or pyramid pooling, yet their efficient transfer of information between encoder and decoder stages proves advantageous for the task at hand, particularly in detecting oil spills and distinguishing challenging classes like "Ships" and "Look-Alike."

In contrast, PSPNet achieves the lowest mean IoU (0.5810) despite offering the fastest inference time (2.21 ms per image). CBD-Net, which is adapted for multi-class segmentation in this study, achieves a mean IoU of 0.6088. This falls slightly below expectations, likely due to the challenges of adapting a binary classification model to the more complex multi-class task.

The inference times of the models also offer valuable insights into their computational performance. PSPNet and DeepLabv3+ stand out with the fastest inference times, with PSPNet achieving the best result (2.21 ms) at the cost of lower accuracy. In contrast, U-Net and LinkNet show slower inference times (9.05 ms and 9.21 ms, respectively), which can be attributed to their computationally intensive upsampling and concatenation operations in the decoder stages.

Class	U-Net	LinkNet	PSPNet	DeepLabv3+	CBD-Net
Sea Surface	0.9555	0.9527	0.9327	**0.9609**	0.9437
Oil Spill	**0.5554**	0.5485	0.4444	0.5262	0.5261
Look-alike	**0.4918**	0.4247	0.3652	0.4874	0.4461
Ships	0.2672	**0.3405**	0.2141	0.3131	0.2588
Land	**0.9500**	0.9326	0.9485	0.9392	0.8696
Mean IoU	0.6440	0.6398	0.5810	**0.6454**	0.6088

Table 2: Testing Results

While these processes enhance accuracy, they increase computational demand. DeepLabv3+ strikes a valuable balance, achieving high accuracy alongside a relatively fast inference time of 5.47 ms. This positions it as the most effective model for the task when considering the trade-off between accuracy and computational requirements.

Considering both accuracy and inference time, DeepLabv3+ emerges as the optimal solution for the problem at hand.

4.3 Qualitative analysis

Figure 4 illustrates the segmentation performance of the five models on three examples from the oil spill dataset. Each row depicts different input images, while the columns show the segmentation output of a model.

U-Net and LinkNet both perform well in delineating large regions of oil spills (cyan) and land (green), with U-Net benefiting from skip connections and LinkNet leveraging residual connections. These design features preserve boundary details and support segmentation of small objects like ships (brown). However, both models struggle in differentiating oil spills from look-alike regions (red) when grayscale intensities are similar, leading to occasional false positives or blurred boundaries. LinkNet slightly outperforms U-Net on ships, reflecting its strength in small-object segmentation.

PSPNet, while being the fastest model in terms of inference time, produces the least accurate segmentations. It struggles with thin or complex features, often merging narrow oil spills with adjacent classes. This under-segmentation is evident in Figure 4, where PSPNet fails to preserve the fine boundaries of spill regions.

DeepLabv3+, which achieves the highest mean IoU, shows a more conservative behaviour towards oil spill segmentation, reducing false positives and ensuring balanced segmentation between oil spills and look-alike areas. Its multi-scale strategy preserves fine boundaries, resulting in outputs that closely align with the ground truth across all examples. This is especially noticeable in challenging scenarios such as dark grayscale images, where other models like PSPNet and CBD-Net tend to over-segment look-alike regions. As illustrated in the first row of Figure 4, DeepLabv3+ demonstrates a clear advantage in accurately distinguishing oil spills from dark regions, further solidifying its position as the best-performing model.

CBD-Net captures the overall shape of large oil spills, benefiting from its contextual and boundary refinement design. However, adapting it from binary to multi-class segmentation introduces challenges, such as overextending spill boundaries or misclassifying small features like ships.

Overall, these results confirm DeepLabv3+ as the most reliable model for accurate and balanced segmentation across all classes, with a clear advantage in challenging dark-image scenarios.

4.4 Extensions results

To explore ways of improving model training, the two alternative training strategies are implemented: sliding window and focused crop, explained in section 3.2.1. These techniques are tested on the DeepLabv3+ architecture, which proved to be the most promising in the earlier experiments.

While the sliding window approach is often effective in scenarios with limited data, the results produced here are suboptimal. Since the systematic cropping of images led to a dataset with 8 times as many datapoints as the original one, the training for this experiment is only carried out for 200 epochs. The trained sliding window model achieves a mean IoU of 0.5726, significantly lower than the 0.6453 mean IoU obtained with random cropping after 600 epochs. Performance on minority classes such as “Oil Spill” and “Look-Alike” is particularly poor. One possible explanation for the poorer performance could reside in the lack of positional variations produced by the sliding window approach, where cropped patches are fixed and repetitive. This might limit the model’s ability to generalize to unseen data, as it is exposed to fewer diverse perspectives. In contrast, resizing and random cropping over the whole images introduces greater variability, potentially improving generalization by forcing the model to learn from different perspectives of the same region. Further investigation into hybrid approaches or more sophisticated augmentations could provide insights into optimizing the sliding window method.

On the other hand, the focused crop method produces more promising results. Although this method does not lead to improved overall performance, as reflected in the final IoU scores, it demonstrates significant benefits in terms of training convergence. Figure 5 shows that the model trained with focused crop reaches satisfactory performance on oil spills approximately 50 epochs earlier than the baseline with random cropping. This faster convergence suggests that focused crop introduces more relevant features earlier in training, potentially making it a valuable strategy for reducing training time in future work. Refining the cropping logic further, such as optimizing the probability or offset distribution, may yield even better results.

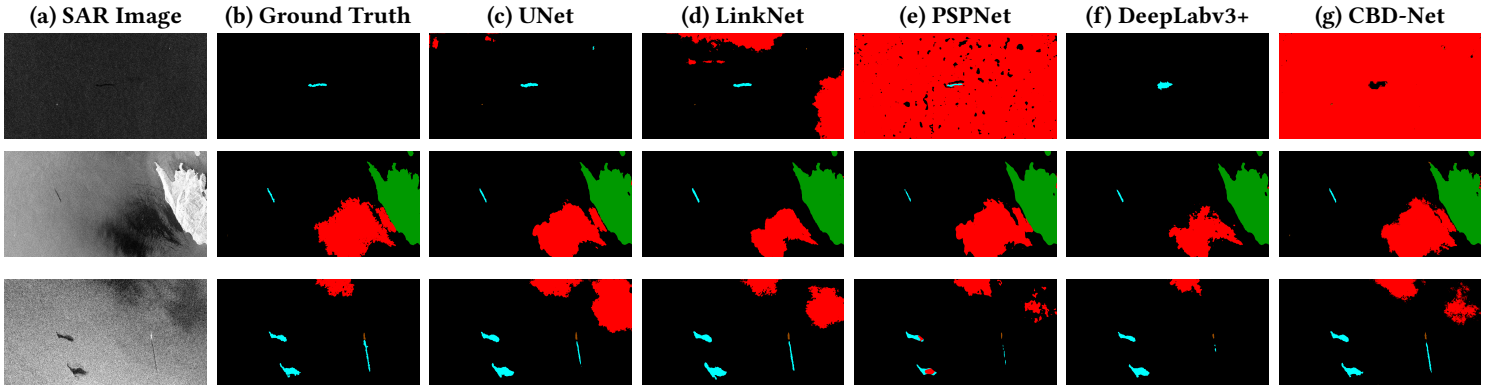

Figure 4: Qualitative results of the examined segmentation models on the presented oil spill dataset. Columns correspond to (a) SAR Image, (b) Ground Truth, (c) UNet, (d) LinkNet, (e) PSPNet, (f) DeepLabv3+, and (g) CBD-Net. Rows represent different examples.

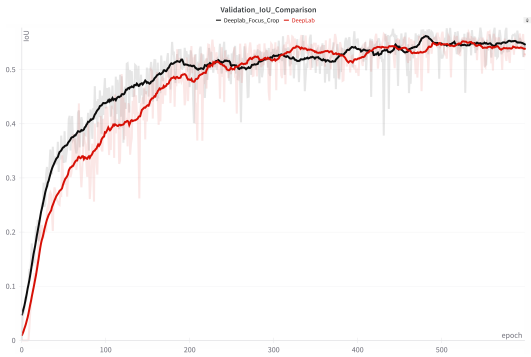

Figure 5: Oil spill IoU on the validation set during DeepLabv3+ training, comparing the focused crop (black) and random crop (red) approaches. The focused crop technique accelerates convergence, achieving comparable oil spill performance approximately 50 epochs earlier.

Table 3 compares the results of both techniques with the baseline random crop approach. Overall, the focused crop method emerges as the more promising extension, offering a meaningful improvement in training efficiency without compromising final performance.

Class	DeepLabv3+	Focused Crop	Sliding Window
Sea Surface	**0.9609**	0.9555	0.9400
Oil Spill	**0.5262**	0.4971	0.4539
Look-alike	**0.4874**	0.4748	0.3277
Ships	0.3131	**0.3288**	0.2551
Land	**0.9392**	0.9217	0.8863
Mean IoU	**0.6454**	0.6356	0.5726

Table 3: Final extensions results

5 Conclusion

This project explores the task of oil spill segmentation on SAR imagery, emphasizing both the potential for efficient and accurate detection and the inherent challenges of processing such data. Through an experimental evaluation of five segmentation models, the study provides a comparative analysis of their architectures and performance. Despite differences in design, all models encountere similar challenges, particularly in segmenting the underrepresented ship class. While the models demonstrate promising results in identifying oil spill regions, significant gaps remain before such approaches can be reliably deployed in real-world scenarios.

Future work could focus on several key areas to enhance the performance and applicability of segmentation models for SAR imagery. One promising avenue is the development of advanced noise reduction techniques tailored specifically for SAR data, such as despeckling techniques, which could improve feature extraction and overall segmentation accuracy [15] [16] [17]. Another direction involves exploring data augmentation and synthetic data generation methods [18] to address class imbalance issues, particularly for underrepresented categories like ships. Research into hybrid approaches that combine more traditional architectures like CNNs with newer solutions, like transformers, may also provide more robust solutions for challenging scenarios [19]. Lastly, a deeper investigation into lightweight and computationally efficient models, like DeepLabv3+, could facilitate real-time deployment in operational settings [20], such as oil spill monitoring systems aboard satellites or drones.

References

[1] Konstantinos Karantzalos, Demetre Argialas. Automatic detection and tracking of oil spills in sar imagery with level set segmentation. *Remote Sensing Laboratory, School of Rural and Surveying Engineering, National Technical University of Athens*, 2008.

[2] Georgios Orfanidis, Konstantinos Ioannidis, Konstantinos Avgerinakis, Stefanos Vrochidis, Ioannis Kompatsiaris. A deep neural network for oil spill semantic segmentation in sar images. *Centre for Research and Technology Hellas (CERTH)-Information Technologies Institute (ITI)*, 2018.

[3] Camilla Brekkea, Anne H.S. Solbergb. Oil spill detection by satellite remote sensing. *Norwegian Defence Research Establishment, Postboks 25, 2027 Kjeller, Norway b, Department of Informatics, University of Oslo, Postboks 1080 Blindern, 0316 Oslo, Norway*, 2004.

[4] Konstantinos N. Topouzelis. Oil spill detection by sar images: Dark formation detection, feature extraction and classification algorithms. *Joint Research Centre (JRC), European Commission, Via Fermi 2749, 21027, Ispra (VA), Italy*, 2008.

[5] Philipp Krahenbuhl, Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Computer Science Department Stanford University*, 2012.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. 2018.

[7] Marios Krestenitis, Georgios Orfanidis, Konstantinos Ioannidis, Konstantinos Avgerinakis, Stefanos Vrochidis and Ioannis Kompatsiaris. Oil spill identification from satellite images using deep neural networks. *Centre for Research and Technology Hellas, Information Technologies Institute, 6th km Harilaou-Thermi, 57001 Thessaloniki, Greece*, 2019.

[8] Marios Krestenitis, Georgios Orfanidis, Konstantinos Ioannidis, Konstantinos Avgerinakis, Stefanos Vrochidis, and Ioannis Kompatsiaris. Early identification of oil spills in satellite images using deep cnns. *Centre for Research Technology Hellas, Information Techologies Institute, Thessaloniki, Greece*, 2019.

[9] Suman Singha, Tim J. Bellerby, and Olaf Trieschmann. Satellite oil spill detection using artificial neural networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2013.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany*, 2015.

[11] Abhishek Chaurasia, Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. *School of Electrical and Computer Engineering and Weldon School of Biomedical Engineering, Purdue University West Lafayette, USA*, 2015.

[12] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang1 Jiaya Jia1. Pyramid scene parsing network. *The Chinese University of Hong Kong*, 2017.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 2017.

[14] Yanan Zhang, Qiqi Zhu, Qingfeng Guan. Oil spill detection based on cbd-net using marine sar image. *School of Geography and Information Engineering, China University of Geosciences, Wuhan, China*, 2021.

[15] Prabhishek Singh, Manoj Diwakar, Achyut Shankar, Raj Shree, and Manoj Kumar. A review on sar image and its despeckling. *Archives of Computational Methods in Engineering*, 28:4633–4653, 2021.

[16] Jie Li, Liupeng Lin, Mange He, Jiang He, Qiangqiang Yuan, and Huanfeng Shen. Sentinel-1 dual-polarization sar images despeckling network based on unsupervised learning. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[17] Malsha V Perera, Wele Gedara Chaminda Bandara, Jeya Maria Jose Valanarasu, and Vishal M Patel. Transformer-based sar image despeckling. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 751–754. IEEE, 2022.

[18] Snyoll Oghim, Youngjae Kim, Hyochoong Bang, Deoksu Lim, and Junyoung Ko. Sar image generation method using dh-gan for automatic target recognition. *Sensors*, 24(2), 2024.

[19] Jonggu Kang, Chansu Yang, Jonghyuk Yi, and Yangwon Lee. Detection of marine oil spill from planetscope images using cnn and transformer models. *Journal of Marine Science and Engineering*, 12(11), 2024.

[20] Jiahao Zhang, Pengju Yang, and Xincheng Ren. Detection of oil spill in sar image using an improved deeplabv3+. *Sensors*, 24(17), 2024.